Bounds Computation for Symmetric Nets

Benoît Barbot, Marco Beccuti, Giuliana Franceschinis, Serge Haddad, Claudine Picaronny

Bounds Computation for Symmetric Nets

Benoît Barbot[1], Marco Beccuti[2], Giuliana Franceschinis[3], Serge Haddad[4⋆], Claudine Picaronny[4]

[1] LACL, Université Paris-Est Créteil, France
E-mail: `benoit.barbot@u-pec.fr`
[2] Dipartimento di Informatica, Università di Torino, Italy
E-mail: `beccuti@di.unito.it`
[3] DiSIT, Università del Piemonte Orientale, Italy
E-mail: `giuliana.franceschinis@di.unipmn.it`
[4] LSV, ENS Paris-Saclay, CNRS, Inria, Université Paris-Saclay France
E-mail: `{haddad,picaronny}@lsv.fr`

Wednesday 7th March, 2018 17:10

Abstract. Monotonicity in Markov chains is the starting point for quantitative abstraction of complex probabilistic systems leading to (upper or lower) bounds for probabilities and mean values relevant to their analysis. While numerous case studies exist in the literature, there is no generic model for which monotonicity is directly derived from its structure. Here we propose such a model and formalize it as a subclass of Stochastic Symmetric (Petri) Nets (SSNs) called Stochastic Monotonic SNs (SMSNs). On this subclass the monotonicity is proven by coupling arguments that can be applied on an abstract description of the state (symbolic marking). Our class includes both process synchronizations and resource sharings and can be extended to model open or cyclic closed systems. Automatic methods for transforming a non monotonic system into a monotonic one matching the MSN pattern, or for transforming a monotonic system with large state space into one with reduced state space are presented. We illustrate the interest of the proposed method by expressing standard monotonic models and modelling a flexible manufacturing system case study.

1 Introduction

Analysis of stochastic models. Bounding models are used to analyze systems with large state spaces when the properties of interest cannot be computed either numerically due the size of the system or statistically due to the rare event problem or difficulties to estimate steady state probabilities. Bounding models are built with additional constraints that make the bounding model lumpable, yielding a smaller state space. Numerical methods are applied on the bounding model to compute an upper or a lower bound of the value of interest.

Applications. Stochastic bounds were first applied in the context of telecommunications networks. Among many other works, methodological approaches have

⋆ The work of this author was partly supported by ERC project EQualIS (FP7-308087).

been proposed in [15] while more specific ones related to the quality of service [4] or message losses [10] have been developed. More recently, stochastic bounds have been designed for the analysis of Web services [9,12].

Our contributions. In the present paper we propose a structural characterization of a class of Stochastic Symmetric (Petri) Nets (SSN) [6,5], called Stochastic Monotonic Symmetric Nets (SMSN), for which a coupling relation exists between abstract states (symbolic markings) and can be exploited to prove its monotonicity. The structure of SMSNs is defined in terms of a precise pattern comprising an alternating sequence of *interfaces* and *zones* through which entities (processes) can flow. Processes may synchronize within zones, moreover both interfaces and zones may have finite capacity. Monotonicity is ensured by defining proper constraints on the way in which the processes can move forward and backward within zones and between zones and interfaces, and on the rates of transitions. The practical interest of this characterization comes from the possibility of automatically transforming a wider class of SSNs, called Pre-Monotonic SNs, into a SMSN from which bounds on some performance indices (e.g. the time to absorption into some final state) of the original model can be computed efficiently by lumping similar states.

The SMSN formalism allows one to define systems that start in an initial state, with all processes ready in the first interface, and end in a (unique) final state. In the paper we show that it is also possible to extend this approach to work with open nets and with cyclic closed nets, by adapting the coupling relation and the performance indices to be bounded.

Related work. Bounding models are classically used to analyze a Markov chain with a large state space. In [1,8,13] algorithms computing bounding models from the transition probability matrix of a Markov chain are presented. These algorithms take also as input an equivalence relation over the state space, states in this relation are aggregated in the bounding model. In [16] a stochastic system is defined as a tensor product of several Markov chains yielding a compact representation for a large Markov chain. These representations are used (for example in [11]) to build bounding models by analyzing each component instead of the whole system.

However, these approaches do not consider formalisms for which monotonicity is guaranteed by construction nor propose any automatic procedure that can transform a non monotonic and only partially symmetric model into a new one satisfying the required properties on which bounds can be efficiently computed.

Outline. In Section 2, two intuitive motivating examples are proposed. In Section 3, we recall the formalism of Stochastic Symmetric nets and illustrate it through a Flexible Manufacturing System (FMS) model. In Section 4, we introduce Monotonic Symmetric nets, summarize coupling theory and establish the monotonicity of MSNs through coupling of (abstract) states. In Section 5, we apply the previous result to obtain bounds of a class of SSNs called Pre-Monotonic, which can be automatically transformed into a bounding SMSN model. In Section 6, we show how the approach may be extended to both open and cyclic

Fig. 1. On the left: a tandem queue. On the right: a tandem queue with capacity.

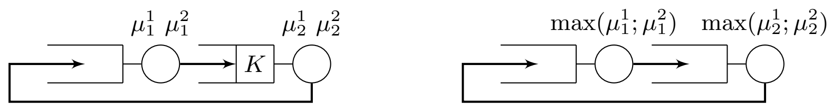


Fig. 2. On the left: a multi-class network with capacity. On the right: a mono-class network.

closed models. Finally in Section 7, we conclude and give some perspectives on this work. In Appendix, the FMS model is fully described.

2 Motivating examples

In order to introduce our approach, we describe here two standard (simple) examples that can be automatically handled in our framework.

Bounding the probability of buffer overflow. A tandem queue (presented on the left of Figure 1) consists in a system where clients enter with some rate (here λ) and then successively wait in two queues to be served. The rates of the services are μ_1 and μ_2. A critical issue for the design of such systems is the size of buffers associated with the current clients. For instance, suppose that the designer wants to know the probability of a buffer overflow between two idle periods when the global number of buffers is B. Then the size of the state space of the corresponding Markov chain is $\Theta(B^2)$. If B is too large prohibiting an exact computation, this probability can be upper bounded by the tandem queue on the right where the capacity of the second queue is K. When the capacity is reached, the server of the first queue is stopped. The size of the state space of the corresponding Markov chain belongs to $\Theta(KB)$ and thus K can be tuned in order to obtain a good trade-off between the computational cost and the accuracy of the bound.

Bounding the throughput. On the left of Figure 2, a closed and two-class version of the tandem queue is represented where a fixed number of clients n_i for $i \in \{1, 2\}$ visit the two queues with service rates depending on the class of clients (featuring also a capacity K for the second queue). Suppose the first queue represents idleness of the clients, then the infinite-server semantic for the service discipline is appropriate. The second queue corresponds to an activity on some server and thus queuing discipline could be FIFO and service discipline could be single server. With such hypotheses, the size of the state space of the corresponding Markov chain is now exponential w.r.t. $n_1 + n_2$. Here we are interested in the throughput of the system, i.e. the number of clients served in some queue per time unit. The queuing system presented on the right of Figure 2 is a mono-class version without capacity restriction whose size of the state space of the corresponding Markov chain is now polynomial w.r.t. $n_1 + n_2$. It can been shown that the throughput of the second system is an upper bound of the one of the first system.

Discussion. For a reader unfamiliar with stochastic ordering, our claims about the bounds seem straightforward. In fact it requires some technical machinery whose main ingredients are: (1) designing some mapping between states of the two systems and (2) establishing that the "bounding system" is *monotonic* in some appropriate way. The framework that we propose avoids to the designer these manual steps and furthermore allows it to tune (as in the first example) the trade-off between accuracy and computational cost.

3 Stochastic symmetric nets

3.1 Preliminary definitions and notations

Before introducing stochastic symmetric nets some preliminary definitions are needed.

Definition 1 (Multiset). *A multiset a over a nonempty set A is a mapping $a \in \mathbb{N}^A$ and $Bag(A)$ is the set of multisets over A.*

Intuitively, a multiset is a set that can contain several occurrences of the same element. It can be represented by a formal sum: $a = \sum_{x \in A}(a(x))x$ The coefficient $a(x)$ is called multiplicity of x in a. A multiset b is smaller than a, denoted $b \sqsubseteq a$, if for all $x \in A$, $b(x) \leq a(x)$.

Definition 2 (Operations on Multisets). *Let $a, b \in Bag(A)$, $n \in \mathbb{N}$. Addition, substraction and scalar multiplication of multisets are defined as follows:*

- $a + b = \sum_{x \in A}(a(x) + b(x))x$;
- *when* $b \sqsubseteq a$, $a - b = \sum_{x \in A}(a(x) - b(x))x$;
- $n \cdot a = \sum_{x \in A} na(x)x$.

Given a family of sets $\{A_i\}_{i=1}^n$, $A_1 \times \cdots \times A_n$ is the Cartesian product of these sets. An item of $A_1 \times \cdots \times A_n$ is denoted $\langle x_1, \ldots, x_n \rangle$ where $x_i \in A_i$. For all $i \in \mathbb{N}$, let $a_i \in Bag(A_i)$. The multiset $\langle a_1, \ldots, a_n \rangle \in Bag(A_1 \times \cdots \times A_n)$ is defined by:

$$\forall i \leq n \; \forall x_i \in A_i \; \langle a_1, \ldots, a_n \rangle(x_1, \ldots, x_n) = a_1(x_1) \cdots a_n(x_n)$$

3.2 Stochastic symmetric nets: an introduction

Petri Nets (PN) and its generalizations (e.g. Generalized Stochastic Petri Nets (GSPN), Colored Petri Net (CPN), Stochastic Symmetric Nets (SSN), etc.) are appropriate formalisms for modelling and analyzing many systems like communication networks, computer systems and manufacturing systems. Petri Nets are bipartite directed graphs with two types of nodes: *places* and *transitions*. The places, graphically represented as circles, correspond to the state variables of the system, while the transitions, graphically represented as rectangles, correspond to the events that trigger state changes. The arcs connecting places to transitions (and vice versa) express the relations between states and event occurrences. Places contain tokens drawn as black dots within the places. The state of a PN, called *marking*, is defined by the number of tokens in each place. A transition is *enabled* if every input place of the transition contains a number of tokens greater than or equal to a given threshold labelling the corresponding input arc. A transition occurrence, called *firing*, removes these tokens from its input places and adds tokens to its output places according to the label of its output arcs.

GSPNs extend PNs with timing specifications introducing two types of transitions: timed and immediate ones. When enabled, a timed transition fires after a random delay specified by a negative exponential probability distribution whose rate may depend on the state. Immediate transitions fire in zero time. The partition of transitions into timed and immediate ones, induces a partition of the states in *tangible* ones (where the system spends time) and *vanishing* ones (where the system does not spend time). The semantic of a GSPN is a Continuous Time Markov Chain (CTMC) representing its underlying stochastic process. The states of the CTMC correspond to the tangible states of the GSPN and the transition rates can be derived from the information contained in the model reachability graph. Hence, the standard analysis of GSPNs consists in the computation of their transient or steady-state probability distribution which can be used to assess classical performance indices.

CPNs extend PNs with the possibility to associate information, called *color*, with tokens and with transition firings, hence defining *firing instances* of a transition producing different state changes. Thus a *color domain*, denoted *cd*, is associated with places and transitions. The enabling condition and the state change associated with each transition instance are specified by means of functions labelling arcs: given the color identifying an instance of the transition connected to the arc, the function provides the multiset of colored tokens that will be added-to or removed-from the place connected to the output or input arc. Thus CNPs are more compact and parametric models.

Similarly to CPNs, SSNs are a high level Petri net formalism which extend GSPNs with colors. Moreover, thanks to a well-structured color syntax, SSNs provide an efficient solution technique which automatically exploits the model symmetries to derive a lumped CTMC reducing the computational cost of the analysis.

Color domains in SSNs are expressed by Cartesian products of *color classes*. $\mathcal{C} = \{C_1, \ldots, C_n\}$ is the set of these classes including the *null product* (ϵ) consisting of a neutral color as in ordinary GSPNs. Color classes may be considered as primitive domains and may be partitioned into *static subclasses*. Intuitively the colors of a class represent entities of the same nature (e.g. raw parts), but only the colors within a static subclass are guaranteed to behave similarly (e.g. raw parts requiring the same manufacturing process). The arc expression corresponding to an arc connecting place p and transition t denotes a function $f : cd(t) \rightarrow Bag(cd(p))$ and is expressed as a (integer) weighted sum of tuples, whose elements are class functions. There are few types of class functions (projection, successor, diffusion/synchronization), allowing to exploit behavioral symmetries in the derivation of an aggregate state space.

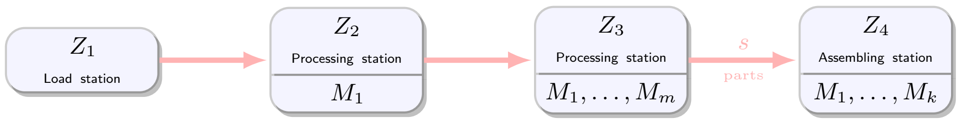


Fig. 3. Flexible Manufacturing System: general schema.

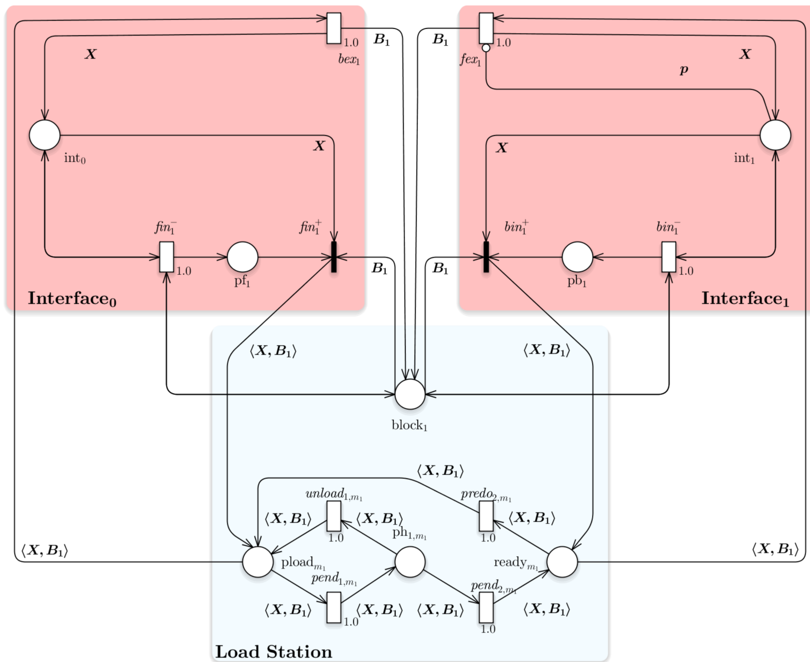
Example 1. An example of SSN model is shown in Figures 4 and and 5 (see also Figures 26 and 27 in the appendix), representing a Flexible Manufacturing System (FMS), a production system consisting of a set of identical and/or complementary numerically controlled machines which are connected through an automated transportation system. The submodels depicted in the four figures share some common place ($\{int_i\}_{i\leq 4}$): by *glueing* them on the common places one gets the complete model; in all submodels a light blue box highlights the portion representing the actual machines processing parts, while the red boxes highlight the interfaces towards the preceding and following machine (showing a structure which is similar in the four submodels). This common structure shall be explained in Section 4.

The modelled FMS, whose general schema is shown in Figure 3, comprises four zones, visited sequentially by the parts to be worked: zone Z_1 contains the load station, zones Z_2 and Z_3 contain processing stations, finally zone Z_4 contains one assembly station. Each processing station is composed of a set of machines that can process the parts circulating in the FMS: a single machine is available in Z_2, m machines are available in Z_3 and k machines are available in Z_4. We assume that all these machines require three phases to complete their task, and at the end of any phase the partially processed parts must pass a quality control before accessing the next phase. A partially processed part that does not pass the quality control must re-start in the first phase. The machines in the assembly station take in input s parts to be assembled in a final product. This task requires four phases, and the partially processed parts must pass a quality control at the end of each phase again.

The following five color classes are defined in the SSN model:

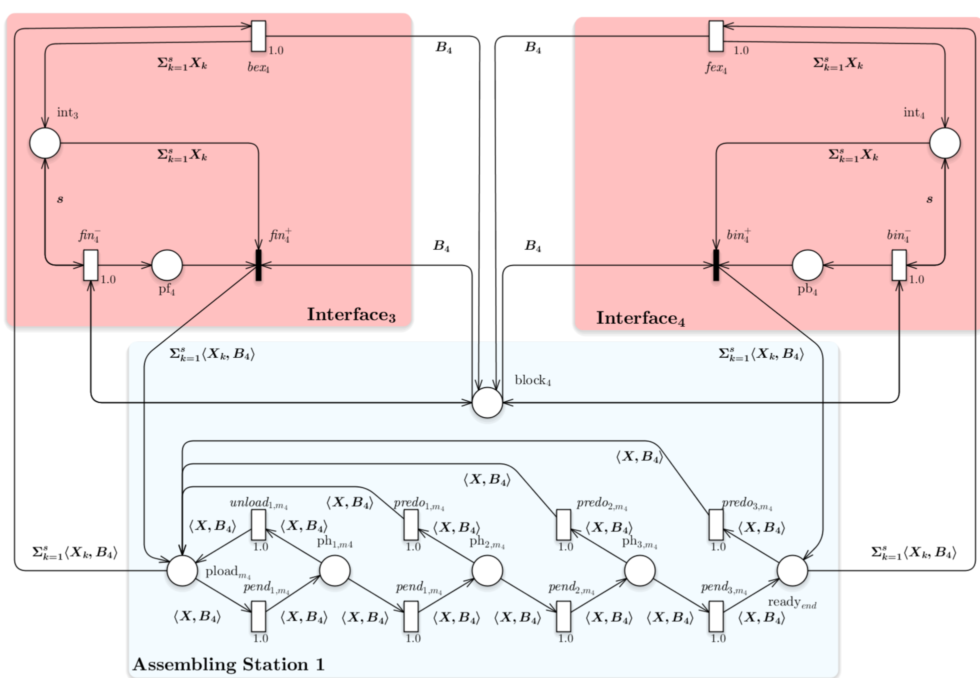
- $Parts = \{c_1, \ldots, c_j\}$ modelling the parts circulating in the FMS;
- $Mach_1 = \{m_{1,1}\}$, $Mach_2 = \{m_{2,1}, \ldots, m_{2,l}\}$, $Mach_3 = \{m_{3,1}, \ldots, m_{3,k}\}$ and $Mach_4 = \{m_{4,1}, \ldots, m_{4,m}\}$ modelling the machines in each zone.

Places Pf_i and Pb_i have a neutral domain, while Places int_i have color domain $cd(int_i) = Parts$, and Places $block_i$ have color domain $cd(block_i) = Mach_i$. All the



Initial marking: $m_0 = All.int_0 + All.block_1$

Fig. 4. The load station of the FMS.



Initial marking: $m_0 = All.block_4$

Fig. 5. The assembling station (with m machines) of the FMS.

other places in zone i have color domain $Parts \times Mach_i$. The color domain of the transitions $pendi_{m0}$ in Figure 4 is $Part \times Mach_1$ where X (ranging over $Part$) and B_1 (ranging over $Mach_1$) are the variables appearing in the functions annotating its input and output arcs. Similarly, the color domain of the transitions fin_3^+ in Figure 5 is $Part^s \times Mach_1$ where $\{X_i\}_{i\leq s}$ (ranging over $Part$) and B_4 (ranging over $Mach_4$) are the variables appearing in the functions annotating its input and output arcs. Constraints can be defined on the transition color domain through guards: boolean expressions whose terms are basic predicates on the transition variables. A *transition instance* is defined as a binding of its variables to actual values in the corresponding class: a valid instance must satisfy the transition guard.

The input and output arcs of each transition are annotated with expressions denoting functions from the transition color domain to multisets on the place color domain. Arc expressions are expressed as weighted sums of variable tuples (e.g. $\sum_{k=1}^{s}\langle X_k, B_i\rangle$); input arc expressions are denoted $Pre(p,t)$, output arc expressions are denoted $Post(p,t)$. For any transition t the expressions on its input (output) places are denoted as a place-indexed sum of arc expressions: $\sum_{p\in {}^\bullet t} Pre(p,t).p \left(\sum_{p\in t^\bullet} Post(p,t).p\right)$.

In details, Figure 4 reports the SSN sub-model describing the first FMS zone, which contains the load station machine and its input and output buffers. To make the figure clearer and readable a blue box was used to highlight the sub-net modelling the load station, and two red boxes to highlight the input and output buffers (respectively $\mathrm{Interface}_0$ and $\mathrm{Interface}_1$).

In the left red box place int_0 models the input buffer of the load station machine. A raw part is assigned to an available machine (contained in place $block_1$ in the blue box) firing the transition sequence fin_1^- and fin_1^+ [5]. An assigned part (i.e. place $pload_{m_0}$ in the blue box) is then processed by its associated machine, which has to be correctly positioned and oriented. Initially, the machine tries to fetch the part: if this task fails then the part is unloaded into the input buffer (i.e. transition bex_1), otherwise the positioning phase is executed (i.e. transition $pend_{1,m_0}$). At the end of this first phase the correct position of this part is verified: if it is not correctly positioned then this phase is repeated (i.e. transition $unload_{1,m_0}$), otherwise the part is ready for the orientation phase. The orientation phase is modeled by transition $pend_{2,m_0}$. A part correctly oriented is directly moved to the output buffer (i.e. place int_1 in the right red box) by transition fex_1, otherwise it is moved back to the first phase by transition $predo_{2,m_0}$. The size limitation of this output buffer is achieved by connecting place int_1 to transition fex_1 with an inhibitor arc (graphically represented with an arc ending with a circle) labelled with the buffer size (i.e. p). In this way, transition fex_1 is enabled and can fire if the total number of tokens in place int_1 is lower than p. Moreover, when a part is inserted in the output buffer its associated machine becomes idle: its corresponding colored token is moved by transition fex_1 into place $block_1$.

[5] The arcs connecting places int_0 and $block_1$ to transition fin_1^- are *read arcs* (i.e. predicates) used to check if the number of tokens in these places (whatever the color) is greater than or equal to 1

All the parts in the output buffer are waiting for being processed by the first processing station, however this processing station may require a re-positioning and re-orientation of any part (i.e. transitions bin_1^- and bin_1^+). Finally, the initial marking for this sub-model assumes all the parts in the input buffer (i.e. $All.int_0$ meaning that place int_0 contains on token per element of its color domain) and the single load station machine in its idle status (i.e. $All.block_0$).

The sub-model in Figure 5 describes instead the fourth FMS zone which contains the assembly station (i.e. sub-net in blue box) and its input and output buffers (i.e. sub-nets in the red boxes). Since the assembly station takes as input s parts to be assembled this synchronization is encoded in the model by the arc function $\sum_{k=1}^{s} X_k$ (resp. $\sum_{k=1}^{s}\langle X_k, B_4\rangle$) labelling the arcs connecting place int_3 to transition fin_4^+, place int_4 to transition bin_4^+, transition bex_4 to place int_3, transition fex_4 to place int_4 (resp. transition fin_4^+ to place $pload_{m4}$, transition bin_4^+ to place $pload_{m4}$, place $pload_{m4}$ to transition bex_4, and place $ready_{end}$ to transition fex_4. The four phases of the assembly process are modeled by places $\{ph_{i,m_4}\}_{i\leq 3}$ and transitions $\{pend_{i,m_4}\}_{i\leq 3}$ and $\{predo_{i,m_4}\}_{i\leq 3}$. The initial marking for this model assumes all the processing station machines in their idle status (i.e. $All.block_4$).

The description of the sub-models for zones two and three, and more details on how the whole SSN model is derived by the sub-models are reported in Appendix A.

3.3 Syntax and semantics

In this section we define (a simplified form of) the Symmetric Nets formalism and provide its semantics.

Definition 3 (Symmetric Net). *An SN is a tuple*

$$\mathcal{N} = (P, T, \mathcal{C}, \Sigma, cd, \mathbf{prio}, label, \mathbf{Guard}, \mathbf{Pre}, \mathbf{Post}, m_0)$$

where:

- P *and* T *are the set of places and transitions respectively;*
- $\mathcal{C} = \{C_i\}_{i=1}^{n}$ *is the set of basic color classes, that may be partitioned into static subclasses denoted* $C_{i,j}$*;*
- Σ *is a set of labels, including a special label* ε*.*
- $cd(p)$ *is the color domain of place* p *defined as the Cartesian product of basic color classes;* $cd(t)$ *is the color domain of a transition, defining its instances, it is expressed as a tuple of variables, e.g.* X_i, Y_i*, whose type is a basic color class* C_i*;*
- $\mathbf{prio} : T \to \mathbb{N}$*, is the priority of transitions;*
- $label : T \to \Sigma$ *is the transition labelling function;* $\mathbf{prio}(t) > 0 \Rightarrow label(t) = \varepsilon$*.*
- **Guard** *associates a guard with each transition; the guard is a boolean function (the default guard is constant function true) denoted through a boolean expression whose terms are* standard predicates *in the form* $X_i = Y_i$ *or* $X_i \in C_{i,j}$ *or* $\#p$ *op* n *where* $p \in P$*, op is a comparison operator,* $n \in \mathbb{N}$*.*

- **Pre** *and* **Post** *associate with each transition t the set of input and output arcs respectively, with the corresponding arc expressions (denoted as a place-indexed sum of arc expressions). An arc expression is a weighted sum of variable tuples (the number and type of variables in the tuples must match the place color domain). An expression on an arc connecting place p and transition t denotes a function $cd(t) \rightarrow Bag(cd(p))$*
- m_0 *is the initial marking, such that for all $p \in P,\ m_0(p) \in Bag(cd(p))$. A marking is denoted by a formal sum whose terms are expressed in the form* 'multiset'.'place name'.

Observe that we are considering a slight extension of guards w.r.t. the original definition of the formalism by adding the possibility of introducing predicates checking whether the number of tokens in a place (whatever the color) is greater than or equal to (resp. less than) a given integer value; graphically this is represented by a special annotation on a *read arc* (i.e. a double headed arc) (resp. an *inhibitor arc* (a circle headed arc)).

Another extension that will be used later in the paper is the addition of a set of labels Σ, including a special label ε, and a labelling function *label* associating a label with each transition of the model; all transitions (with $\mathbf{prio}(t) > 0$) are labelled ε (meaning that they are unobservable).

Instead the arc expression syntax definition is less general than in the original formalism since the tuples can contain only variables (called *projection functions*), while in the more general case they could contain also other elements corresponding to the diffusion/synchronization function, or the successor function.

The semantics of a SN model defines how its state (marking) can evolve from the initial one m_0 to a (possibly infinite) set of *reachable markings*, through sequences of *transition instance firings*. Hence the dynamics of an SN model is defined through the enabling and firing rules. The enabling rule specifies the set of enabled *transition instances* in a given marking m. An instance of transition t is defined through a binding b of the variables in $cd(t)$) to colors in the corresponding classes. An enabled transition instance may fire producing a state change.

The semantics of a transition guard **Guard** is defined as follows: term $X_i = Y_i$ appearing in a guard expression of transition t is true for binding b if the values associated with X_i and Y_i in b are equal; term $X_i \in C_{i,j}$ is true is the value associated with X_i belongs to static subclass $C_{i,j}$. The special term $\#p\ op\ n, p \in P$ is evaluated on a marking m, rather than on a binding, and it is true if $|m(p)|\ op\ n$ (number of tokens contained in p, whatever the colors, is in relation op with n.)

The semantics of arc expressions $\mathbf{Pre}(p,t)$ and $\mathbf{Post}(p,t)$ are defined as follows: the evaluation of a tuple $\langle X_{i1}, \ldots, X_{in} \rangle$ appearing in $\mathbf{Pre}(p,t)$ for binding b of t is a multiset in $Bag(cd(p))$ (where $cd(p) = C_{i1} \times \ldots \times C_{in}$) of cardinality one, containing a single tuple obtained by substituting each variable with the corresponding value in b. The value of the whole expression ($\mathbb{N}$-weighted sum of variable tuples) derives by applying the multiset sum and scalar product semantics.

Definition 4 (Transition instance enabling and firing). *A transition instance* (t, b) *is* enabled *in marking* m *if:*

- $\mathbf{Guard}(t)(b, m) = true$
- $\forall p,\ \mathbf{Pre}(p, t)(b) \sqsubseteq m(p)$
- *no higher priority transition instance* $(t', b') : \mathbf{prio}(t') > \mathbf{prio}(t)$ *satisfies the first two enabling conditions.*

When enabled, the firing *of* (t, b) *from marking* m *leads to marking* m'*, denoted* $m \stackrel{(t,b)}{\rightarrow} m'$*, and defined by: for all* $p \in P$*,* $m'(p) = m(p) - \mathbf{Pre}(p, t)(b) + \mathbf{Post}(p, t)(b)$*.*

Example 2. Let us consider an example of marking and firing in the FMS example:
$m_0 = All.int_0 + All.block_1 + All.block_2 + All.block_3 + All.block_4$
where $All.p$ means that p contains all elements in $cd(p)$.

In m_0 there is an enabled instance of transition fin_1^- (which has neutral color domain): when it fires a *vanishing* marking m_1 is reached defined by:
$m_1 = All.int_0 + 1.pin_0 + All.block_1 + All.block_2 + All.block_3 + All.block_4$.
The presence of a (neutral) token in place pin_0 enables $|Parts|$ instances of immediate transition fin_1^+: $\{(fin_1^+, X = c_i, B_1 = m_{1,1})\}_{i \leq j}$ (for brevity denoted $(fin_1^+, c_i, m_{1,1})$ hereafter).
If $(fin_1^+, c_2, m_{1,1})$ fires from m_1, one reaches m_2 a *tangible* marking defined by:
$m_2 = (All - c_2).int_0 + (\langle c_2, m_{1,1}\rangle).pload_{m0} + All.block_2 + All.block_3 + All.block_4$
Observe that if instance $(fin_1^+, c_5, m_{1,1})$ fires instead, one reaches a quite "similar" marking m_3 defined by:
$m_3 = (All - c_5).int_0 + (\langle c_5, m_{1,1}\rangle).pload_{m0} + All.block_2 + All.block_3 + All.block_4$

From marking m_2 several firing sequences are possible like: $(pend_{1,m_1}, c_2, m_{1,1})$ $(pend_{2,m_1}, c_2, m_{1,1})$ $(fex_1, c_2, m_{1,1})$ leading to the marking m_4 defined by:
$m_3 = (All - c2).int_0 + All.block_1 + (\langle p2\rangle).int_1 + All.block_2 + All.block_3 + All.block_4$
enabling transitions fin_1^-, bin_1^- and fin_2^-.

3.4 Symbolic marking and the SRG

It has been established that due to the particular syntax of the SN formalism two *similar* markings like m_2 and m_3 generate equivalent behaviors, hence one may replace them by a representative *symbolic marking* $\widehat{m}$ and one may define a *symbolic firing rule* to fire *symbolic transition instances* leading to a *symbolic reachability graph* (SRG). On a SRG, most qualitative properties (e.g. existence of deadlock states) can be directly checked. A symbolic marking is an equivalence class of ordinary markings that can be obtained one from the other applying a permutation of colors within static subclasses.

A canonical representation for symbolic markings has been defined in [5], however in the context of this paper we propose a simplified and more intuitive representation illustrated by the FMS example.

Example 3. A possible representation matching the common pattern of markings m_2 and m_3 is:

$$\widehat{m} = (All - x).int_0 + (\langle x, y\rangle).pload_{m0} + All.block_2 + All.block_3 + All.block_4$$

where $x \in Parts, y \in Mach1$. This symbolic marking represents $|Parts|$ equivalent markings, obtained by assigning a specific color c_i to the placeholder x and machine $m_{1,1}$ (unique color in $Mach1$) to the placeholder y. There is one enabled transition instance in any marking represented by $\widehat{m}$, denoted $(pend_{1,m_1}, X = x, B_1 = y)$. The symbolic marking reached after the firing of this transition is:

$$\widehat{m}' = (All - x).int_0 + (\langle x, y\rangle).ph1_{m0} + All.block_2 + All.block_3 + All.block_4$$

with $x \in Parts, y \in Mach1$. From here several symbolic transition instance sequences are possible. After firing $(pend_{2,m_1}, x, y)$ and then (fex_1, x, y) the following symbolic marking is reached:

$$\widehat{m}'' = (All - x).int_0 + All.block_1 + (\langle x\rangle).int_1 + All.block_2 + All.block_3 + All.block_4$$

In this symbolic marking there are three enabled symbolic transition instances fin_0^-, bin_1^- and fin_1^-. If the third one is fired a vanishing symbolic marking is reached, enabling a symbolic immediate transition (fin_1^+, x, y) representing $|Mach2|$ ordinary instances (since y represents any of the $|Mach2|$ colors in class $Mach2$).

Given an initial symbolic marking $\widehat{m}_0$ (in our example the initial symbolic marking includes only one ordinary marking m_0: indeed any permutation of colors within static subclasses maps m_0 on itself), the set of reachable symbolic marking (SRS - Symbolic reachability set) is the smallest set satisfying:

1. $\widehat{m}_0 \in SRS$
2. $\widehat{m} \in SRS \wedge \widehat{m} \xrightarrow{(t,\hat{b})} \widehat{m}' \Rightarrow \widehat{m}' \in SRS$

where $(t, \hat{b})$ denotes a symbolic firing. The SRG is a graph whose set of nodes is the SRS and there is an arc with label $(t, \hat{b})$ between node $\widehat{m}$ and $\widehat{m}'$ iff $\widehat{m} \xrightarrow{(t,\hat{b})} \widehat{m}'$.

3.5 Stochastic SN

Let us introduce additional information to our SN model, required to specify its stochastic timed behavior: each timed transition instance has an associated random delay which is exponentially distributed with a given rate parameter ρ (the rate may depend on the specific binding, but in a constrained way so that behavioral symmetries are preserved): the possible conflict between timed transitions is resolved according to a race policy. Each immediate transition fires in zero time after enabling, and the conflict between immediate transitions with same priorities is solved by a random choice derived from the weights of the transitions.

Definition 5. *A Stochastic Symmetric Net (SSN) is a symmetric net whose transitions can be timed or immediate and have weigths w.*

- $T = T_T \cup T_I$, *where* $T_T = \{t \in T : prio(t) = 0\}, T_I = \{t \in T : prio(t) > 0\}$;
- $w_t : cd(t) \times \bigotimes_{p \in P} Bag(cd(p)) \to \mathbb{R}$ *is interpreted as a rate of a negative exponential distribution if* $t \in T_T$ *or as a weight to be normalized to obtain a firing probability if* $t \in T_I$ *(needed for conflict resolution among immediate transitions with same priority). It is defined as a composition of two functions* $g_t \circ f_t$ *where* f_t *is a* symmetric counting function $f_t : cd(t) \times \bigotimes_{p \in P} Bag(cd(p)) \to \mathbb{N}^k$ *and* $g_t : \mathbb{N}^k \to \mathbb{R}$ *returns a rate, based on the* k counters *derived by* f_t.

In the above definition, f_t is required to be symmetric: this means that for any *permutation* ξ of colors within the model static subclasses, for any given binding b and marking m: $f_t(\xi.b, \xi.m) = f_t(b, m)$. Due to this property the stochastic process of a SSN is a Continuous Time Markov Chain (CTMC) that can be derived from its symbolic reachability graph.

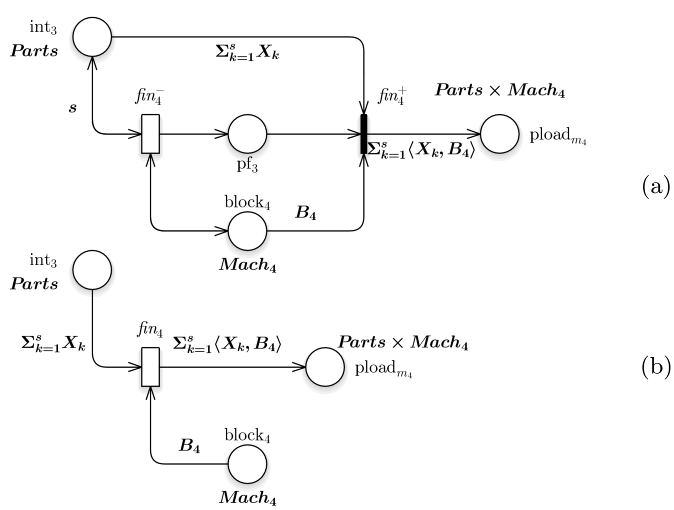


Fig. 6. Examples of color and marking dependent rates

Example 4. The above definition of function w_t allows to model several service policies for transitions in SSNs, and to make a transition rate depend both on the color and (in a constrained way) on the marking. Let us consider for example the SWN depicted in Figure 6(a), which is a portion of the submodel in Figure 5: the firing rate for transition fin_4^- could be proportional to the number of combination of parts that are ready to enter zone 4 from interface int_3. Since the color domain of this transition is neutral, there can be only one instance; it is enabled whenever $\#int_3 \geq s$. Hence in this case it would be appropriate to define $f_{fin_4^-}(b, m) = \lfloor \frac{\#int_3}{s} \rfloor$ and $g_{fin_4^-} = \lambda f_{fin_4^-}$. Let us now consider immediate transition fin_3^+, there can be some enabled instance of this transition in marking m when there is one token in place pf_3, at least s tokens in int_3, and at least one token in $block_4$ in marking m. If place int_3 contains more than s tokens, or place $block_4$ contains more than one token, then there will be several (conflicting) instances of such transition. Let us assume $f_{fin_4^+}(b, m) = 1$, and $g_{fin_4^+} = 1.0\ f_{fin_4^+}$: one among the enabled instances is selected with uniform probability. This simply means that the actual identities of the group of synchronizing parts and the identity of the machine identity associated with the group are randomly chosen. Observe that the future behavior of the FMS model will be identical up to a permutation of parts and machine colors, so that the actual choice is irrelevant from the point of view of computation of performance indices that do not explicitly refer to parts and machine identities.

By exploiting marking dependent rates, it could be possible to eliminate immediate transition fin_4^+: let us consider the submodel in Figure 6(b). The goal is to define the rate of this transition in such a way that it represents as many activities in parallel as the number of blocks ready to synchronize $\lfloor \frac{\#int_3}{s} \rfloor$. On the other hand observe that now the transition is not neutral as it was in the previous example, instead there are several enabled instances depending on the number of tokens in int_3 and in $block_4$ (as discussed earlier for the immediate transition fin_3^+). The presence of several enabled instances, if not controlled through proper definition of the rate, may artificially increase the actual rate for the activity modelled by this transition.

The following tricky definition of functions $f_{fin_4^+} \in \mathbb{N}^2$ and $g_{fin_4^+}$ obtains the same result already achieved in the previous example:

$$f_{fin_4^+} = (\lfloor \frac{\#int_3}{s} \rfloor, (\#block_4 \prod_{k=1}^{s} (\#int_3 - k + 1))$$

and $g_{fin_4^+} = \lambda f_{fin_4^+}[1]/f_{fin_4^+}[2]$. where the denominator of the fraction in the formula counts the number of enabled instances of fin_4^+ (assuming that places $block_4$ and int_3 contain only tokens with different colors).

As discussed before the semantic of a SSN model is a CTMC. So for formalizing this semantic, we first recall the definition of a Markov chain.

Definition 6 (Continuous Time Markov Chain). *A CTMC is a pair* $(S, \boldsymbol{Q})$ *where:*

- S *is the set of states, including an initial state* s_0*;*
- $\boldsymbol{Q}$ *is an* $S \times S$ *matrix of non negative reals called the infinitesimal generator, where* $s, s' \in S, s \neq s', \boldsymbol{Q}(s, s')$ *corresponds to the transition rate between states* s *and* s'*, and* $\forall s \in S$, $\boldsymbol{Q}(s, s) = -\sum_{s' \neq s} \boldsymbol{Q}(s, s')$

As for GSPNs (symbolic) markings can be partitioned into *vanishing* and *tangible* markings: the model does not spend time in the former type of markings, while it spends time in the latter. The vanishing markings can be eliminated by substituting vanishing paths in the reachability graph with direct arcs, so that the result is a reduced RG containing only tangible markings. This structure is isomorphic to a CTMC whose states are the tangible markings, and have a transition from state m to state m' if there is at least one timed transition instance (possibly followed by a sequence of immediate transition instances) whose firing leads from m to m'. The rate of the transition from m to m' in the CTMC is equal to the sum of the rates of all transitions leading from m to m' (computed using the weight function w_t) in the RG.

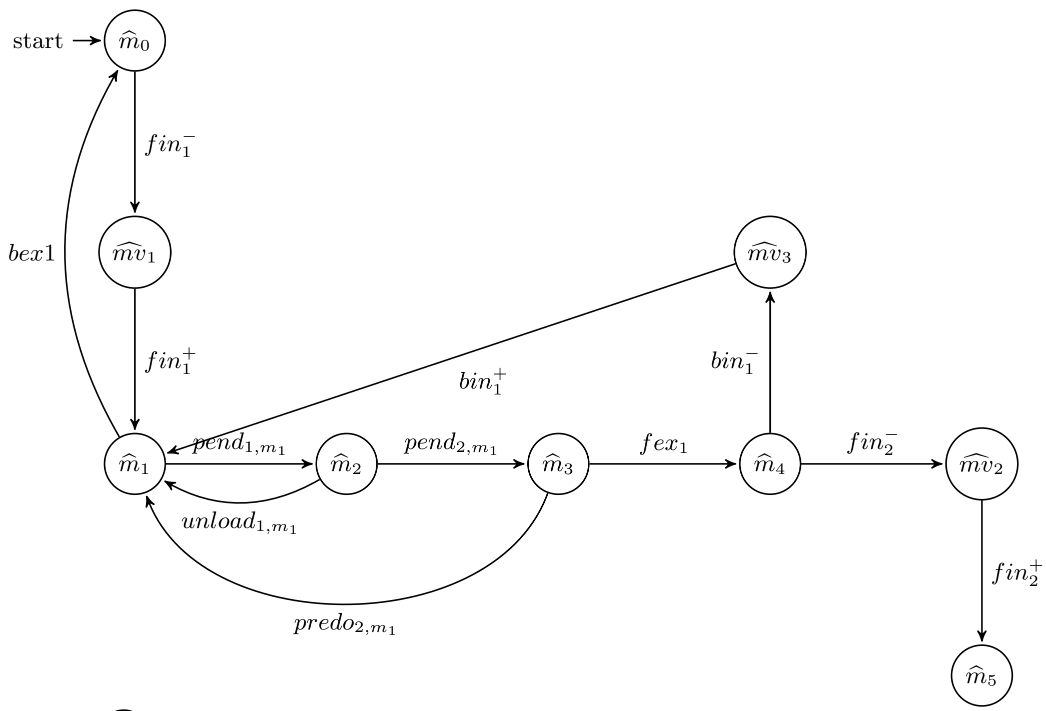
As detailled in [5] the symbolic marking defines a partition of ordinary markings into equivalence classes that satisfy the strong and exact lumpability conditions for CTMC, and it is possible to directly derive a lumped CTMC from the symbolic RG without need to build the complete RG first.

Example 5. Let us consider again a few possible traces of execution of the FMS model presented in Example 3 illustrated in the upper part of Figure 7 and discuss how they translate in corresponding paths in the underlying CTMC, shown in the bottom part of the same figure. The states of the CTMC are in one-to-one relation with the tangible symbolic markings of the SSN: all vanishing paths are hence reduced and substituted by simple arcs in the underlying CTMC. In Figure 7 state s_i of the CTMC corresponds to state $\widehat{m}_i$ in the SRG.

The initial marking $\widehat{m}_0$ is symmetric, meaning that it represents just one ordinary marking, and from it, by firing transition fin_1^- (whose color domain is the neutral one), it is possible to reach symbolic marking $\widehat{mv}_1$ which is also symmetric. The rate of this transition instance depends on the number of tokens in place int_0, denoted $\#int_0$, which in this case is $|Parts|$. The only enabled transition is now the immediate transition fin_1^+: there is only one symbolic binding associated with it (corresponding to several ordinary ones), and its firing leads to tangible marking $\widehat{m}_1$ with probability 1.0. This is the only path leading from $\widehat{m}_0$ to $\widehat{m}_1$, and in fact it is reduced to an arc from s_0 to s_1 in the CTMC, with rate $\lambda_1|Parts|$.

From $\widehat{m}_1$ there are two enabled symbolic instances bex_1 and $pend1, m_1$, both representing a single ordinary firing; the former leads back to $\widehat{m}_0$ with rate λ_0 while the latter leads to $\widehat{m}_2$ with rate λ_2. This directly translates in a transition from s_1 to s_2 in the underlying CTMC.

Similarly from $\widehat{m}_2$ and $\widehat{m}_3$ there are two enabled instances each leading directly to a tangible marking. Finally from $\widehat{m}_4$ there are two possible ways out, one corresponding to the sequence bin_1^-, bin_1^+, reduced to a single arc from s_4 to s_1 with rate μ_5 in the CTMC, and another one corresponding to the sequence fin_2^-, fin_2^+, reduced to a single arc from s_4 to s_5 with rate λ_5 in the CTMC.



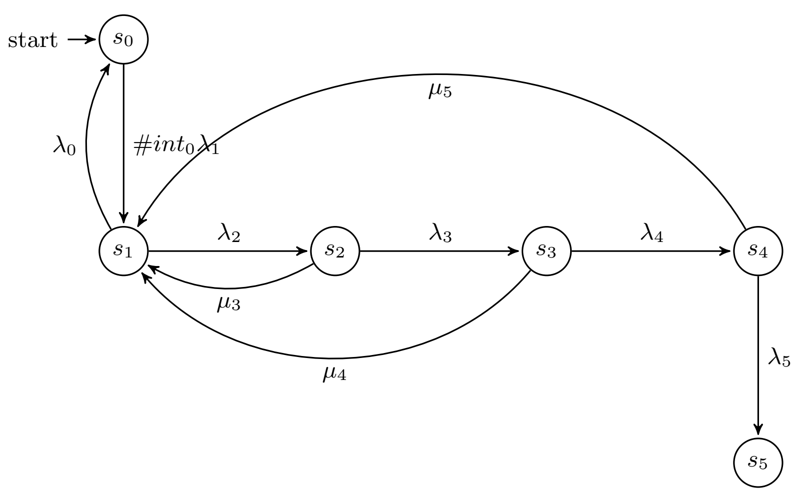


Fig. 7. A portion of SRG and the corresponding CTMC

4 Monotonic symmetric nets

This section presents the core of our contribution: a subclass of symmetric nets for which a coupling relation between symbolic markings can be established. We first introduce the syntax and semantics of this subclass that we illustrate by the example of section 3. Then we recall the basics of coupling and finally we define a binary relation between symbolic markings showing that it is a coupling relation.

4.1 Syntax

We consider a particular case of symmetric nets, called *monotonic symmetric nets*. These nets model processes that interact by synchronization and/or by resource sharing. Such processes perform sequentially a set of state-based *tasks* indexed by $\{1, \ldots, \zeta\}$. Thus the first class of these symmetric nets is $Proc$. In Example 1, the processes are the parts to be processed.

Every task i is executed inside a *zone* and requires a *block* of s_i processes to be done. Several tasks may be concurrently executed inside but there is a possible restriction on the number of simultaneous blocks ($r_i \in \mathbb{N} \cup \{\infty\}$). In order to keep track of the synchronization, every process of the block is paired with a color in a dedicated class $Sync_i$. Inside a zone i, processes evolve without synchronization from local states to local states (denoted by places $p_{i,j}$) where n_i is the number of states. The local states are in some sense ordered: so $p_{i,1}$ (resp. p_{i,n_i}) is the initial (resp. final) state of the zone. An internal transition from local state $p_{i,j}$ is denoted by $t_{i,j,a}$ (where a is the label of the transition) and $p_{i,\delta(i,j,a)}$ is the new local state. In Example 1, there are four zones and the machines are modelled by the synchronization items: for instance in the second zone, $r_2 = 1$. There are s parts to be processed in parallel in the forth zone.

In order to go from a zone to another zone, processes must sojourn in *interfaces* indexed by $\{0, \ldots, \zeta\}$ where interface 0 is the initial state of the processes, interface i, with $0 < i < \zeta$, lies between zones i and $i + 1$, and interface ζ is the final state of the processes. As for zones, the number of processes in an interface may be limited by a capacity ($c_i \in \mathbb{N} \cup \{\infty\}$). In Example 1, the interfaces consist in buffers where the parts wait to be processed and interface 1 has a finite capacity (p).

A process enters interface i either by exiting zone i using transition fex_i (a forward exit) or by exiting zone $i + 1$ using transition bex_{i+1} (a backward exit). In the former case, it requires that all processes of a block are in the final state of zone i while in the latter all processes of a block are in the initial state of zone $i + 1$.

A process enters zone $i + 1$ (resp. i) from interface i by a sequence of a timed transition fin^-_{i+1} (resp. bin^-_i) followed by an immediate transition fin^+_{i+1} (resp. bin^+_i). The timed transition is enabled when there are enough processes to form a block and there is still room for another block. The immediate transition selects the processes for the block and transfer them into the zone.

In order to get a coupling relation (see subsection 4.4), we partition the label of transitions Σ into *forward* labels Σ_f and *backward* labels Σ_b. An internal transition $t_{i,j,a}$ with $a \in \Sigma_f$, moves a process "forward", i.e. $\delta(i,j,a) > j$. Furthermore from any local state $p_{i,j'}$ with j' between j and $\delta(i,j,a)$, there must be a transition $t_{i,j',a}$ with $\delta(i,j',a) \geq \delta(i,j,a)$. A symmetric condition holds for backward labels. In Example 1, the "redo" transitions of zone 4 are backward transitions that from any local state bring back a part to the initial local state of the zone.

Definition 7 (Monotonic Symmetric Net). *Let* $\zeta \in \mathbb{N}$, $(n_i)_{1\leq i\leq\zeta}, (s_i)_{1\leq i\leq\zeta} \in \mathbb{N}^\zeta$, $(r_i)_{1\leq i\leq\zeta} \in (\mathbb{N}\cup\{\infty\})^\zeta$ *and* $(c_i)_{1\leq i\leq\zeta} \in (\mathbb{N}\cup\{\infty\})^{\zeta+1}$. *A monotonic symmetric net* $\mathcal{N} = (\Sigma, \mathcal{C}, \delta, P, T, \mathbf{prio}, cd, \mathbf{Guard}, \mathbf{Pre}, \mathbf{Post}, m_0)$ *is defined by:*

- $\Sigma = \Sigma_f \uplus \Sigma_b$ *a finite alphabet partitioned into forward events* Σ_f *and backward events* Σ_b*;*
- $\mathcal{C} = \{Proc\} \cup \{Sync_i\}_{1\leq i\leq\zeta}$ *with* $Proc = \{1,\ldots,n\}$ *and* $Sync_i = \{1,\ldots,r_i\}$*. We denote the variables associated with Proc,* $X, X_1, \ldots$ *and the variable associated with* $Sync_i$*,* B_i*;*
- δ *is a* partial *function from* $\{(i,j,a) \mid 1 \leq i \leq \zeta, 1 \leq j \leq n_i, a \in \Sigma\}$ *to* $\mathbb{N}$*, with all* $n_i > 1$*, such that:*
  1. *for all* $a \in \Sigma_f$*, (when defined)* $j \leq \delta(i,j,a) \leq n_i$*;*
  2. *for all* $a \in \Sigma_b$*, (when defined)* $1 \leq \delta(i,j,a) \leq j$*;*

  δ *fulfills the* monotonic *conditions: for all* $1 \leq i \leq \zeta$*, for all* $1 \leq j < j' \leq n_i$
  1. *for all* $a \in \Sigma_f$*, if* $\delta(i,j,a)$ *is defined and* $\delta(i,j,a) > j'$ *then* $\delta(i,j',a)$ *is defined and* $\delta(i,j',a) \geq \delta(i,j,a)$*;*
  2. *for all* $a \in \Sigma_b$*, if* $\delta(i,j',a)$ *is defined and* $\delta(i,j',a) < j$ *then* $\delta(i,j,a)$ *is defined and* $\delta(i,j,a) \leq \delta(i,j',a)$*.*
- $P = \bigcup_{i=1}^{\zeta}\{p_{i,1},\ldots,p_{i,n_i}\} \cup \{Int_i\}_{0\leq i\leq\zeta} \cup \{Block_i\}_{1\leq i\leq\zeta}$ $\cup\ \{pb_i, pf_i \mid 1 \leq i \leq \zeta\}$*, a finite set of places. One denotes* $P_i = \{p_{i,1},\ldots,p_{i,n_i}\}$*;*
- *For all* $1 \leq i \leq \zeta, p \in P_i$*,* $cd(p) = Proc \times Sync_i$*,* $cd(Block_i) = Sync_i$*,* $cd(pf_i) = cd(pb_i) = \varepsilon$ *and for all* $0 \leq i \leq \zeta, a \in \Sigma$*,* $cd(Int_i) = Proc$*;*
- $T = \{t_{i,j,a} \mid 1 \leq i \leq \zeta, 1 \leq j \leq n_i, a \in \Sigma, \delta(i,j,a)$ *is defined*$\}$ $\cup\ \{fex_i, bex_i, fin_i^-, bin_i^-, fin_i^+, bin_i^+ \mid 1 \leq i \leq \zeta\}$*, a finite set of transitions. All transitions have priority 0 except* $\{fin_i^+, bin_i^+\}$ *which have priority 1.*
- *For all (defined)* $t_{i,j,a}$*,* $cd(t_{i,j,a}) = Proc \times Sync_i$*,* $\mathbf{Pre}(t_{i,j,a}) = \langle X, B_i\rangle \cdot p_{i,j}$ *and* $\mathbf{Post}(t_{i,j,a}) = \langle X, B_i\rangle \cdot p_{i,\delta(i,j,a)}$*;*
- *For all* fin_i^-*,* $cd(fin_i^-) = \varepsilon$*,* $\mathbf{Pre}(fin_i^-) = 0$*,* $\mathbf{Post}(fin_i^-) = pf_i$ *and* $\mathbf{Guard}(fin_i^-) = \#Int_{i-1} \geq s_i \wedge \#Block_i > 0$*;*
- *For all* fin_i^+*,* $cd(fin_i^+) = Proc^{s_i} \times Sync_i$*,* $\mathbf{Pre}(fin_i^+) = B_i \cdot Block_i + pf_i + \sum_{k=1}^{s_i} X_k \cdot Int_{i-1}$ *and* $\mathbf{Post}(fin_i^+) = \sum_{k=1}^{s_i}\langle X_k, B_i\rangle \cdot p_{i,1}$*;*
- *For all (defined)* bin_i^-*,* $cd(bin_i^-) = \varepsilon$*,* $\mathbf{Pre}(bin_i^-) = 0$*,* $\mathbf{Post}(bin_i^-) = pb_i$ *and* $\mathbf{Guard}(bin_i^-) = \#Int_i \geq s_i \wedge \#Block_i > 0$*;*
- *For all (defined)* bin_i^+*,* $cd(bin_i^+) = Proc^{s_i} \times Sync_i$*,* $\mathbf{Pre}(bin_i^+) = B_i \cdot Block_i + pb_i + \sum_{k=1}^{s_i} X_k \cdot Int_i$ *and* $\mathbf{Post}(bin_i^+) = \sum_{k=1}^{s_i}\langle X_k, B_i\rangle \cdot p_{i,n_i}$*;*

- *For all* $1 \leq i \leq \zeta$, $cd(fex_i) = Proc^{s_i} \times Sync_i$, $\mathbf{Pre}(fex_i) = \sum_{k=1}^{s_i} \langle X_k, B_i \rangle \cdot p_{i,n_i}$,
  $\mathbf{Post}(fex_i) = \sum_{k=1}^{s_i} X_k \cdot Int_i + B_i \cdot Block_i$ *and* $\mathbf{Guard}(fex_i) = \#Int_i + s_i \leq c_i$;
- *For all* $1 \leq i \leq \zeta$, $cd(bex_i) = Proc^{s_i} \times Sync_i$, $\mathbf{Pre}(bex_i) = \sum_{k=1}^{s_i} \langle X_k, B_i \rangle \cdot p_{i,1}$,
  $\mathbf{Post}(bex_i) = \sum_{k=1}^{s_i} X_k \cdot Int_{i-1} + B_i \cdot Block_i$ *and* $\mathbf{Guard}(bex_i) = \#Int_{i-1} + s_i \leq c_{i-1}$.
- $m_0 = All.Int_0 + \sum_{1 \leq i \leq \zeta} All.Block_i$.

Observations. We require that $c_0 = \infty$. For sake of readability, the previous definition requires an alternation of interfaces and zones. In fact, a sequence of contiguous interfaces is always possible while a sequence of zones is possible when all but the first zone are not *synchronized* zones (i.e. the corresponding s_i's fulfill $s_i = 1$). Since we allow r_i to be infinite, the colour domain $Sync_i$ may be infinite. However in a reachable marking, all but a finite number of colours of $Sync_i$ only occur in place $Block_i$.

Discussion. A monotonic symmetric net represents a fixed set of processes performing a finite number of tasks. Furthermore in order to achieve their tasks the number of processes n must be a multiple of all s_i. This could be seen as a restriction on the modelling power of this class of nets. In fact, the real applications that we target are mainly the ones described in section 6. However for sake of clarity, we choose to develop the method on the basic pattern and explain later how to adapt it for the extensions.

Notations. In order to define the stochastic features of the net, we introduce some (symmetric) counters related to a marking m. These counters are associated with different numbers of processes:

- $m \cdot int_i$ is the number of processes in interface i;
- $m \cdot geint_i$ is the number of processes in interfaces j for $j \geq i$ and zones j for $j > i$;
- $m \cdot gezone_i$ is the number of processes in interfaces j and zones j for $j \geq i$ (thus $m \cdot gezone_{i+1} = m \cdot geint_i - m \cdot int_i$).

Definition 8 (Stochastic Monotonic Symmetric Net). *A stochastic monotonic symmetric net is a monotonic symmetric net $\mathcal{N}$ where the weight of transitions is defined as follows.*

- *For all immediate transitions $fint_i^+, bint_i^+$, the weight is equal to 1. This entails that processes that constitute a new block are equiprobably chosen among the available ones.*
- *Let $t \in \{t_{i,j,a} \mid 1 \leq i \leq \zeta, 1 \leq j \leq n_i, a \in \Sigma, \delta(i,j,a)$ is defined$\} \cup \{fex_i, bex_i \mid 1 \leq i \leq \zeta\}$. Then the counting vector f_t is defined by the counters $\{geint_i\}_{1 \leq i \leq \zeta}, \{gezone_i\}_{0 \leq i \leq \zeta}$ and g_t is non decreasing (resp. non increasing) function w.r.t. any counter when t is some $t_{i,j,a}$ with $a \in \Sigma_f$ (resp. $a \in \Sigma_b$) or some fex_i (resp. bex_i).*

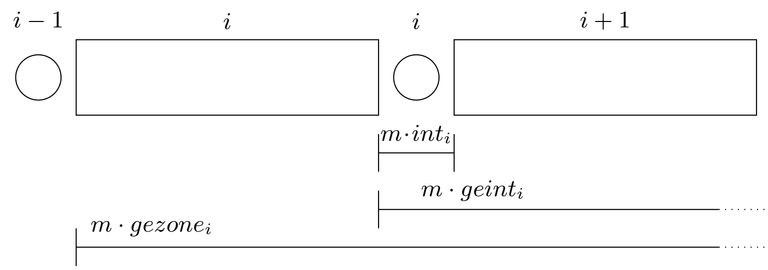


Fig. 8. Illustration of counters for marking m

- *Let* $0 \leq i \leq \zeta$ *and* $t \in \{fin^-_{i+1}, bin^-_i\}$ *(when defined). Then the counting vector* f_t *is defined by the counters* $\{geint_j\}_{1 \leq j \leq \zeta}, \{gezone_j\}_{0 \leq j \leq \zeta}, int_i$ *and* g_t *is non decreasing (resp. non increasing) function w.r.t. any counter when* $t = fin^-_{i+1}$ *(resp.* bin^-_i *).*

Furthermore the weights must fulfill the following requirements.

- *For all* $a \in \Sigma_f$*, if* $\delta(i,j,a)$ *is defined and* $\delta(i,j,a) > j' > j$ *then* $g_{t_{i,j',a}} \geq g_{t_{i,j,a}}$*;*
- *for all* $a \in \Sigma_b$*, if* $\delta(i,j',a)$ *is defined and* $\delta(i,j',a) < j < j'$ *then* $g_{t_{i,j',a}} \leq g_{t_{i,j,a}}$*.*

4.2 Semantics

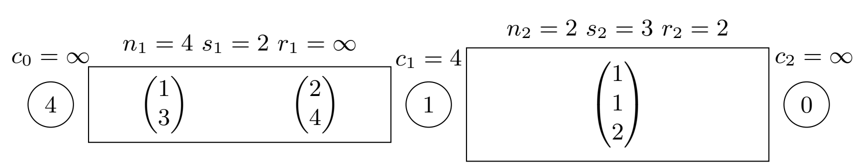


Fig. 9. Symbolic marking representation.

In order to reason about reachable symbolic tangible markings which are the states of the continuous time Markov chain associated with a stochastic monotonic symmetric net, we introduce an appropriate representation for a symbolic marking **Ms** of such nets.

Notations. Let $1 \leq i \leq \zeta$. Then $\mathbf{Vect}_i$ is the subset of $\mathbb{N}^{s_i}$ defined by $\mathbf{Vect}_i = \{(\ell_1, \ldots, \ell_{s_i}) \mid 1 \leq \ell_1 \leq \cdots \leq \ell_{s_i} \leq n_i\}$. In words, $\mathbf{Vect}_i$ are vectors of locations in zone i such that the locations are non decreasing. The partial order $\leq_{s_i}$ on $\mathbf{Vect}_i$ is defined by: $(\ell_1, \ldots, \ell_{s_i}) \leq_{s_i} (\ell'_1, \ldots, \ell'_{s_i})$ if and only if for all j, $\ell_j \leq \ell'_j$.

– Given a block of synchronized processes in the i^{th} zone, we can forget the identity of the synchronization and by ordering their locations get a vector of $\mathbf{Vect}_i$. We define $\mathbf{Ms} \cdot zone_i$ as the multiset of such vectors representing blocks of processes. The set of such multisets is denoted Bag_i. In addition, $|\mathbf{Ms} \cdot zone_i|$ denotes the size of the multiset.
– One can forget the identities of the set of processes in the i^{th} interface and memorize their number denoted by $\mathbf{Ms} \cdot Int_i$.
– We introduce additional useful abbreviations $\mathbf{Ms} \cdot geint_i = \sum_{j \geq i} \mathbf{Ms} \cdot Int_j + \sum_{j>i} s_j |\mathbf{Ms} \cdot zone_j|$ denote the number of processes that have reached the i^{th} interface or beyond. $\mathbf{Ms} \cdot gezone_i = \sum_{j \geq i} \mathbf{Ms} \cdot Int_j + \sum_{j \geq i} s_j |\mathbf{Ms} \cdot zone_j|$ denote the number of processes that have reached the i^{th} zone or beyond.

Example 6. Figure 9 graphically illustrates this representation with two zones represented by rectangles and three interfaces represented by circles. For instance, $\mathbf{Ms} \cdot Int_0 = 4$ and $\mathbf{Ms} \cdot zone_1 = (1, 3) + (2, 4)$. Similarly, $\mathbf{Ms} \cdot geint_1 = 4$ and $\mathbf{Ms} \cdot gezone_2 = 3$.

4.3 Coupling method

The *coupling* method [14] is a classical method for comparing two stochastic processes. It can be applied in various contexts (establishing ergodicity of a chain, stochastic ordering, bounds, etc.). A coupling between two Markov chains is also a Markov chain whose state space is a subset of the product of the two spaces. This subset is called *the coupling relation.* A coupling must satisfy that the projection of a coupling on any of its components must behave like the original corresponding chain.

For our needs, we use Markov chains enriched with events labeling transitions.

Definition 9 (Enriched Markov chain). *An enriched continuous time Markov chain C is a tuple $(S, s_0, \Sigma, \delta, \lambda)$ defined by:*

– *a set of states S including an initial state s_0;*
– *a finite set of events Σ;*
– *a set of labeled transitions $\delta \subset S \times \Sigma \times S$;*
– *a rate function $\lambda : \delta \to \mathbb{R}^+$*

We define the infinitesimal generator matrix $\boldsymbol{Q}$ of size $S \times S$ by:

$$\forall s \neq s' \in S, \boldsymbol{Q}(s, s') = \sum_{(s,e,s') \in \delta} \lambda(s, e, s') \text{ and } \boldsymbol{Q}(s, s) = -\sum_{s' \neq s} \boldsymbol{Q}(s, s')$$

Example 7. An enriched Markov chain is presented in Figure 10. The set of events Σ is defined by $\Sigma = \{a, b\}$. The rate from s_1 to s_2, $\mathbf{Q}(s_1, s_2)$, is defined by $\mathbf{Q}(s_1, s_2) = \nu_a + \mu_b$.

Depending of the context for which the coupling is used, additional constraints are imposed. For our purposes, we provide a coupling relation of an enriched Markov chain with itself such that the time to reach the unique absorbing state s_f from state s' is smaller or equal than the one from state s whenever (s, s') belongs to the coupling relation.

Definition 10. *Let $\mathcal{C} = (S, s_0, \Sigma, \delta, \lambda)$ be an enriched Markov chain with a unique absorbing state $s_f \in S$. A coupling of $\mathcal{C}$ with itself is a* CTMC *$\mathcal{C}^{\otimes} = (S^{\otimes}, (s_0, s_0), \Sigma, \delta^{\otimes}, \lambda^{\otimes})$ such that:*

- $S^{\otimes} \subseteq S \times S$
- $\forall s\ , t\ s'\ , t'\ \in S\ , \forall e, e' \in \Sigma \mid s \neq s'$:
  $\lambda(s, e, t) = \sum_{e' \in \Sigma} \lambda^{\otimes}((s, s'), (e, e'), (t, t'))$ *and*
  $\lambda(s', e', t') = \sum_{e \in \Sigma} \lambda^{\otimes}((s, s'), (e, e'), (t, t'))$
- $\forall (s, s') \in S^{\otimes},\ s = s_f \Rightarrow s' = s_f$

The set $S^{\otimes}$ defines a coupling relation *with a reachability goal s_f.*

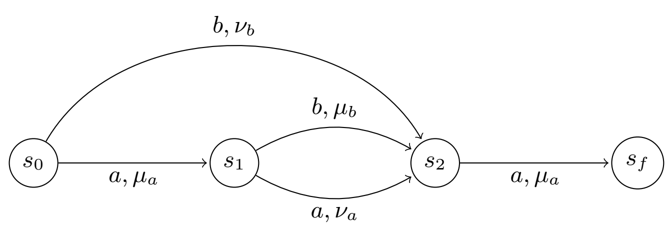


Fig. 10. An enriched CTMC.

Example 8. For the enriched CTMC described by figure 10 with $\nu_a < \mu_a$ and $\nu_b < \mu_b$, it seems that a coupling relation with a reachability goal s_f could be defined by the order relation $s_0 \prec s_1 \prec s_2 \prec s_f$: $\mathcal{S}^{\otimes} = \{(s, t) | s \prec t\}$. However it cannot be directly established. In order to achieve this goal, one adds self-loops corresponding to "missing rates" as illustrated in Figure 11. After adding these self-loops all states have an outgoing rate μ_a (resp. μ_b) for label a (resp. b). Such self-loops can be added without modifying the behaviour of continuous Markov chains. We let the reader check that, with this completion, one can define a chain $\mathcal{C}^{\otimes}$ over $\mathcal{S}^{\otimes}$ that is a coupling relation w.r.t. s_f.

In order to compare the hitting times to the only absorbing state from two coupled states, we need to uniformize the enriched CTMC according to its labels : Let C$= (S, s_0, \Sigma, \delta, \lambda)$ be an enriched continuous time Markov chain. For each event e in Σ, we denote by μ_e the maximum of all $\sum_{s'} \lambda(s, e, s')$ for all states s. We then add for each state s such that $\lambda(s, e) < \mu_e$ a loop (s, e, s) with rate $\mu_e - \sum_{s'} \lambda(s', e)$.

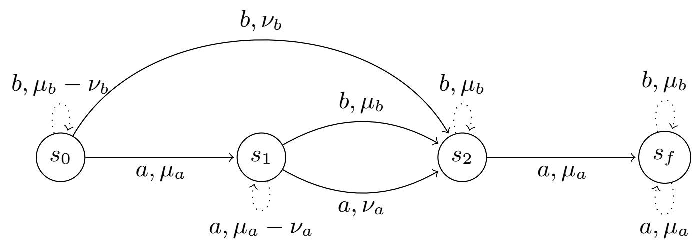


Fig. 11. Completion of the CTMC of Figure 10.

The following proposition allows to compare the hitting time to the final state s_f without any numerical computation. Let us denote $Reach(s, s_f)$ the hitting time to the state s_f in $\mathcal{C}$ starting from state s.

Theorem 1. *Let $\mathcal{C}^{\otimes}$ be a coupling of $\mathcal{C}$, with a reachability goal s_f. Then, for all $(s, s') \in S^{\otimes}$, for all $\tau > 0$, we have:*

$$\mathbb{P}(Reach(s, s_f) \leq \tau) \leq \mathbb{P}(Reach(s', s_f) \leq \tau)$$

Proof.
The unique absorbing state of the chain $S^{\otimes}$ is (s_f, s_f). Let σ be a finite random trajectory ending in (s_f, s_f) starting from (s, s') in $\mathcal{C}^{\otimes}$. As $\forall (t, t') \in S^{\otimes},\ t = s_f \Rightarrow t' = s_f$, we have

$$Reach((s, s'), \{s_f\} \times S)(\sigma) = Reach((s, s'), (s_f, s_f))(\sigma) \geq Reach((s, s'), S \times \{s_f\})(\sigma).$$

Thus, $\mathbb{P}(Reach((s, s'), \{s_f\} \times S) \leq \tau) \leq \mathbb{P}(Reach((s, s'), S \times \{s_f\}) \leq \tau)$
By projection on each component,

$$\begin{aligned} \mathbb{P}(Reach(s, s_f) \leq \tau) &= \mathbb{P}(Reach((s, s'), \{s_f\} \times S) \leq \tau) \\ &\leq \mathbb{P}(Reach((s, s'), S \times \{s_f\}) \leq \tau) \\ &= \mathbb{P}(Reach(s', s_f)) \leq \tau) \end{aligned}$$

$\square$

4.4 Coupling stochastic monotonic symmetric nets

As seen before, a coupling is defined by a binary relation between states of a CTMC, in our case symbolic markings. The intuition underlying this relation is the following one: a symbolic marking is paired with another one if one can match processes associated with the two markings such that for every pair, the process of the former marking is *more or equally advanced* than the process of the latter marking. In fact, the binary relation we define is a partial order.

Definition 11. *Let $\mathcal{N}$ be a* MSN *and* $\mathbf{Ms}, \mathbf{Ms}'$ *be two symbolic reachable markings of $\mathcal{N}$. Then* $\mathbf{Ms} \leq \mathbf{Ms}'$ *if:*

(C1) *for all* $1 \leq i \leq \zeta$, *(C1i)* $\mathbf{Ms} \cdot geint_i \leq \mathbf{Ms}' \cdot geint_i$ *and (C1z)* $\mathbf{Ms} \cdot gezone_i \leq \mathbf{Ms}' \cdot gezone_i$;

(C2) *Let* $\Delta_i = \max(\frac{1}{s_i}(\mathbf{Ms} \cdot gezone_i - \mathbf{Ms}' \cdot geint_i), 0)$ *for* $1 \leq i \leq \zeta$ *(Δ_i is an integer). Then for all* i*, there exists* R_i*, a multiset of pairs of vectors in* $\mathbf{Vect}_i \times \mathbf{Vect}_i$ *such that:*
   1. *For all* $(\mathbf{v}, \mathbf{v}') \sqsubseteq R_i$, $\mathbf{v} \leq_{s_i} \mathbf{v}'$;
   2. $|R_i| = \Delta_i$;
   3. $proj_1(R_i) \sqsubseteq \mathbf{Ms} \cdot zone_i$ *and* $proj_2(R_i) \sqsubseteq \mathbf{Ms}' \cdot zone_i$.

Example 9. Figure 12 shows two ordered symbolic markings. For instance, $\mathbf{Ms} \cdot geint_1 = 4, \mathbf{Ms} \cdot gezone_1 = 8$ and $\mathbf{Ms}' \cdot geint_1 = 6, \mathbf{Ms}' \cdot gezone_1 = 10$. Observe that $\Delta_1 = \frac{1}{2}(8 - 6) = 1$ and that the corresponding pair of vectors fulfills $(1, 3) \leq_2 (2, 3)$.

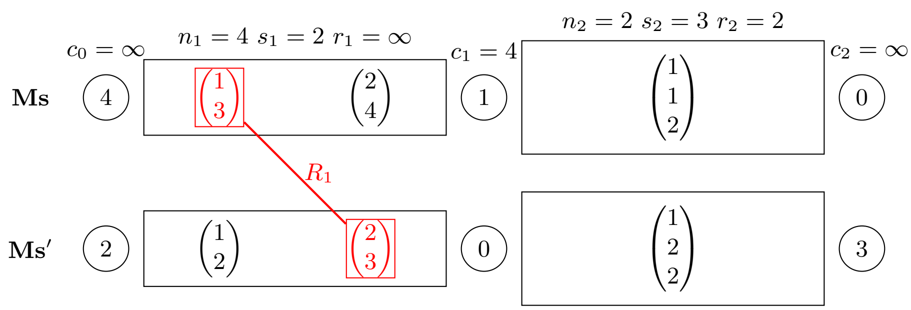


Fig. 12. Comparing symbolic markings.

This definition implies that given a pair $\mathbf{Ms} \leq \mathbf{Ms}'$, if a local/exit forward (resp. backward) transition is simultaneously firable in $\mathbf{Ms}$ and $\mathbf{Ms}'$ then its firing rate in $\mathbf{Ms}'$ is greater (resp. smaller) or equal than its firing rate in $\mathbf{Ms}$. The case of an entry in a zone is more involved (see case 3 of the proof of Theorem 2).

Lemma 1. *Let* $\mathcal{N}$ *be a* MSN *and* $\leq$ *the binary relation of Definition 11. Then* $\leq$ *is a partial order.*

Proof. We first establish antisymmetry. Let $\mathbf{Ms}$ and $\mathbf{Ms}'$ fulfill $\mathbf{Ms} \leq \mathbf{Ms}'$ and $\mathbf{Ms} \geq \mathbf{Ms}'$. So for all $1 \leq i \leq \zeta$, $\mathbf{Ms} \cdot geint_i = \mathbf{Ms}' \cdot geint_i$ and $\mathbf{Ms} \cdot gezone_i = \mathbf{Ms}' \cdot gezone_i$. This implies that in $\mathbf{Ms}$ and $\mathbf{Ms}'$, the number of processes in an interface and the number of blocks in a zone are equal for all interfaces and blocks. This also implies that for all i, Δ_i is equal to the number of blocks in zone i. Consider a maximal vector $\mathbf{v}$ w.r.t $\leq_{s_i}$ over all blocks of zone i occurring in $\mathbf{Ms}$ or $\mathbf{Ms}'$. Let $n_{\mathbf{v}}$ (resp. $n'_{\mathbf{v}}$) be the number of occurrences of $\mathbf{v}$ in $\mathbf{Ms}$ (resp. $\mathbf{Ms}'$). If $n_{\mathbf{v}} > n'_{\mathbf{v}}$ (resp. $n_{\mathbf{v}} > n'_{\mathbf{v}}$) at least one occurrence of $\mathbf{v}$ in $\mathbf{Ms}$ (resp. $\mathbf{Ms}'$) cannot matched with a vector in $\mathbf{Ms}'$ (resp. $\mathbf{Ms}$). So $n_{\mathbf{v}} = n'_{\mathbf{v}}$. Iterating the reasoning over the remaining vectors, one concludes that $\mathbf{Ms} = \mathbf{Ms}'$.

Consider now transitivity. Let $\mathbf{Ms}$, $\mathbf{Ms}'$ and $\mathbf{Ms}''$ fulfill $\mathbf{Ms} \leq \mathbf{Ms}'$ and $\mathbf{Ms}' \leq \mathbf{Ms}''$. Condition $(C1)$ is implied by transitivity of $\leq$ over integers. Let Δ_i (resp. Δ'_i, Δ''_i) be defined as in Definition 11 for pair $(\mathbf{Ms}, \mathbf{Ms}')$ (resp. $(\mathbf{Ms}', \mathbf{Ms}'')$, $(\mathbf{Ms}, \mathbf{Ms}'')$). If $\Delta''_i = 0$, we are done. Otherwise, let $(\mathbf{v}, \mathbf{v}') \sqsubseteq R_i$. If there exists a pair $(\mathbf{v}', \mathbf{v}'') \sqsubseteq R'_i$ (where we assume that identical vectors have some distinguishing "identities") then one adds $(\mathbf{v}, \mathbf{v}'')$ to R''_i. Let us show that R''_i is enough big. $\Delta''_i = \Delta_i - \frac{1}{s_i}(\mathbf{Ms}''.geint_i - \mathbf{Ms}'.geint_i)$. Since $\frac{1}{s_i}(\mathbf{Ms}''.geint_i - \mathbf{Ms}'.geint_i)$, is an upper bound of the number of blocks in $\mathbf{Ms}'$ that do not belong to R''_i. Thus $|R''_i| \geq \Delta''_i$ so that some items can be omitted to reach the desired size[6].

□

Lemma 2. *Let $\mathcal{N}$ be a* MSN *equipped with the order of Definition 11. Let* $\mathbf{Ms}$ *be a symbolic marking that reaches by a forward (resp. backward) move* $\mathbf{Ms}'$ *(resp.* $\mathbf{Ms}''$*). Then* $\mathbf{Ms} \leq \mathbf{Ms}'$ *(resp.* $\mathbf{Ms} \geq \mathbf{Ms}''$*).*

Proof. Since the backward and forward requirements are dual, we only examine forward transitions. Checking condition $(C1)$ is straightforward due to the definition of a forward transition.

- Consider a transition in a zone performed by a block. Then for all i, Δ_i is equal to the number of blocks in zone i for $\mathbf{Ms}$ (and also $\mathbf{Ms}'$). So we match a block with itself and again the definition of a forward transition implies that a block is at least as advanced in $\mathbf{Ms}'$ as in $\mathbf{Ms}$.
- Consider a transition exiting a zone i performed by a block. Then for all $j \neq i$, Δ_j is equal to the number of blocks in zone i for $\mathbf{Ms}$ (and also $\mathbf{Ms}'$). So we match a block with itself. Δ_i is equal to the number of blocks in zone i for $\mathbf{Ms}'$. So we match all the blocks of $\mathbf{Ms}$ except the one that has left zone i with themselves.
- Consider a transition entering a zone i and constituting a new block. Then for all j, Δ_j is equal to the number of blocks in zone i for $\mathbf{Ms}$. So we match a block of $\mathbf{Ms}$ with itself.

□

Theorem 2. *Let $\mathcal{N}$ be a* MSN *equipped with the order of Definition 11. Then this order defines a coupling between symbolic markings of $\mathcal{N}$.*

Proof.

Part one : Matching the processes. Let $\mathbf{Ms}$ and $\mathbf{Ms}'$ be two symbolic markings. We first proceed by establishing a matching between processes in $\mathbf{Ms}$ and $\mathbf{Ms}'$. This matching proceeds inductively from most to least advanced processes.

Basis case. Observe that due to $(C1i)$, $\mathbf{Ms} \cdot Int_\zeta \leq \mathbf{Ms}' \cdot Int_\zeta$. so we match $\mathbf{Ms} \cdot Int_\zeta$ processes and it remains $\mathbf{Ms}' \cdot Int_\zeta - \mathbf{Ms} \cdot Int_\zeta$ unmatched processes in $\mathbf{Ms}'$.

[6] In fact one can prove that $|R''_i| = \Delta''_i$.

Inductive case for interfaces. Assume that we have matched all the processes in $\mathbf{Ms}$ beyond the i^{th} interface so that it remains $\mathbf{Ms}' \cdot gezone_{i+1} - \mathbf{Ms} \cdot gezone_{i+1}$ unmatched processes in $\mathbf{Ms}'$ beyond the i^{th} interface.
Observe that due to $(C1i)$, $\mathbf{Ms} \cdot Int_i \leq \mathbf{Ms}' \cdot Int_i + \mathbf{Ms}' \cdot gezone_{i+1} - \mathbf{Ms} \cdot gezone_{i+1}$. so we match the $\mathbf{Ms} \cdot Int_i$ processes at the i^{th} interface with the unmatched processes of $\mathbf{Ms}'$ and possibly with some processes at the i^{th} interface. and it remains $\mathbf{Ms}' \cdot geint_i - \mathbf{Ms} \cdot geint_i$ unmatched processes in $\mathbf{Ms}'$.

Inductive case for zones. Assume that we have matched all the processes in $\mathbf{Ms}$ beyond the i^{th} zone so that it remains $\mathbf{Ms}' \cdot geint_i - \mathbf{Ms} \cdot geint_i$ unmatched processes in $\mathbf{Ms}'$ beyond the i^{th} zone.
Observe that due to $(C1z)$, $s_i \mathbf{Ms} \cdot zone_i \leq s_i \mathbf{Ms}' \cdot zone_i + \mathbf{Ms}' \cdot geint_i - \mathbf{Ms} \cdot geint_i$. Furthermore, due to the structure of the net, $\mathbf{Ms}' \cdot geint_i - \mathbf{Ms} \cdot geint_i$ is a multiple of s_i. Condition $(C2)$ implies the existence of a multiset of pairs of blocks in the i^{th} zone of size $\Delta_i = \max(\frac{1}{s_i}(\mathbf{Ms}' \cdot geint_i - \mathbf{Ms} \cdot gezone_i), 0)$. If $\Delta_i = 0$ then all blocks of processes in $\mathbf{Ms}$ may be matched with unmatched processes of $\mathbf{Ms}'$ beyond the i^{th} zone. Otherwise, the number of processes in unmatched blocks (by R_i) of the i^{th} zone in $\mathbf{Ms}$ is equal to:

$$s_i \mathbf{Ms} \cdot zone_i + \mathbf{Ms}' \cdot geint_i - \mathbf{Ms} \cdot zone_i = \mathbf{Ms}' \cdot geint_i - \mathbf{Ms} \cdot geint_i$$

which is exactly the unmatched processes in $\mathbf{Ms}'$ beyond the the i^{th} zone. We (arbitrarily) match the processes of the unmatched blocks (by R_i) with these unmatched processes of $\mathbf{Ms}'$ and we match the other processes using R_i. It remains $\mathbf{Ms}' \cdot gezone_i - \mathbf{Ms} \cdot gezone_i$ unmatched processes in $\mathbf{Ms}'$.

There are interesting properties for this matching. First a process in $\mathbf{Ms}$ is always matched by a process at least as advanced. Second when two matching processes are in the same zone, all the processes of the corresponding blocks are matched together.

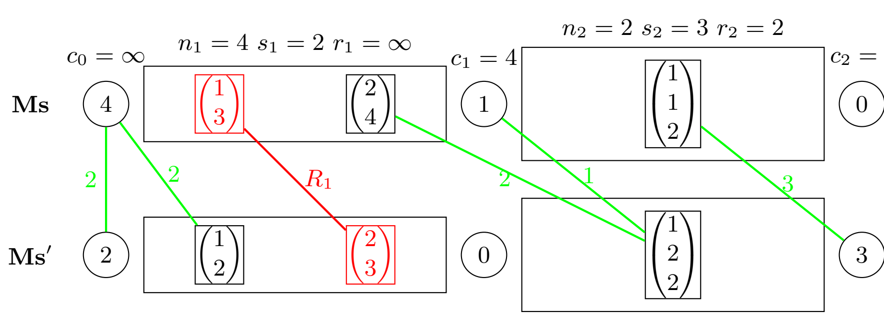


Fig. 13. Matching processes in symbolic markings.

Part two : Matching the transitions. The second part of the proof consists in matching the transitions outgoing from $\mathbf{Ms}$ and $\mathbf{Ms}'$ such that the symbolic markings reached by these transitions are still ordered, introducing self-loops when needed as discussed in the coupling method in order to take into account the rates of transitions. So let $\mathbf{Ms} \xrightarrow{\mu} \mathbf{Ms}_1$ and $\mathbf{Ms}' \xrightarrow{\mu'} \mathbf{Ms}'_1$ be transitions to be matched.

- When $\mu = \mu'$, one must prove that $\mathbf{Ms}_1 \leq \mathbf{Ms}'_1$;
- When $\mu > \mu'$, one must prove that $\mathbf{Ms}_1 \leq \mathbf{Ms}'_1$ and $\mathbf{Ms}_1 \leq \mathbf{Ms}'$;
- When $\mu < \mu'$, one must prove that $\mathbf{Ms}_1 \leq \mathbf{Ms}'_1$ and $\mathbf{Ms} \leq \mathbf{Ms}'_1$.

We will consider transitions triggered by the pairs of processes obtained by the matching. We have to perform a case per case study. Since the backward and forward requirements are dual, we only examine the forward transitions.

Case 1: A transition in a zone. Assume that a process π of $\mathbf{Ms}$ is in location j of zone i and there is a forward transition labelled by a with rate ρ from j to j_1 that leads to symbolic marking $\mathbf{Ms}_1$. Let us consider the matching process π' in $\mathbf{Ms}'$. There are several cases to be examined.

• **Case 1.1: Process π' is beyond zone i.**
If it does not trigger a transition labelled by a then by considering a self-loop with rate ρ, one must show that $\mathbf{Ms}_1 \leq \mathbf{Ms}'$. Condition $(C1)$ still holds since all processes of $\mathbf{Ms}_1$ are in the same zone or interface. Condition $(C2)$ is established by considering the same bags of pairs R_i (observe that the block of the process π in $\mathbf{Ms}$ does not belong to R_i due to the location of its peer in $\mathbf{Ms}'$).
If π' triggers a transition labelled by a with rate ρ' to symbolic marking $\mathbf{Ms}'_1$, then by Lemma 2, $\mathbf{Ms}' \leq \mathbf{Ms}'_1$ and since $\mathbf{Ms}_1 \leq \mathbf{Ms}$, $\mathbf{Ms}' \leq \mathbf{Ms}'_1$ (using Lemma 1). So whatever the relative values of μ and μ', these two relations establish the matching.

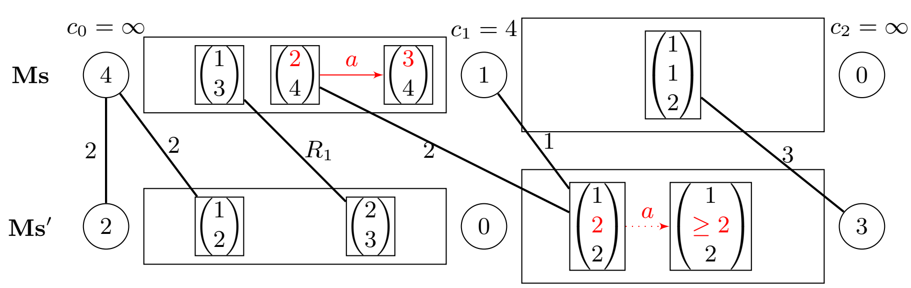


Fig. 14. Illustrating Case 1.1. The color red indicates which process is moving and the dashed arrow points out that the transition may not exist.

• **Case 1.2: Process π' is in zone i in location j'.** This implies that blocks associated with π and π' are matched by relation R_i.
If π' does not trigger a transition labelled by a then by considering a self-loop with rate ρ, one must show that $\mathbf{Ms}_1 \leq \mathbf{Ms}'$. We first observe that our requirements

on nets imply $j' \geq j_1$. Condition $(C1)$ still holds since all processes of $\mathbf{Ms}_1$ are in the same zone or interface. Condition $(C2)$ is established by considering the same bags of pairs R_i. Indeed the new location of π, j_1 is smaller or equal than the the location of π', j' and so their blocks can still be matched.
If π' triggers a transition labelled by a, with rate $\rho' < \rho$ leading to location j'_1, our requirements on nets imply $j' \geq j_1$. By considering a self-loop with rate $\rho - \rho'$, one must show that $\mathbf{Ms}_1 \leq \mathbf{Ms}'$ (already done) and $\mathbf{Ms}_1 \leq \mathbf{Ms}'_1$. Since $j'_1 \geq j'$ the proof is identical to the proof of $\mathbf{Ms}_1 \leq \mathbf{Ms}'$.
If π' triggers a transition labelled by a, with rate $\rho' \geq \rho$ leading to location j'_1, our requirements imply $j'_1 \geq j_1$. By considering a self-loop for $\mathbf{Ms}$ with rate $\rho' - \rho$, one must show that $\mathbf{Ms}_1 \leq \mathbf{Ms}'_1$ (already done) and $\mathbf{Ms} \leq \mathbf{Ms}'_1$. By Lemma 2, $\mathbf{Ms} \leq \mathbf{Ms}_1$ and by transitivity (Lemma 1) $\mathbf{Ms} \leq \mathbf{Ms}'_1$.

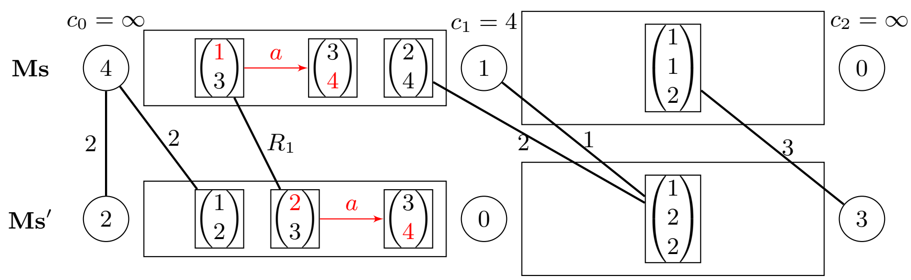


Fig. 15. Illustrating Case 1.2.

Case 2: Exiting a zone. Assume that all processes of a block, say Π, of $\mathbf{Ms}$ are in location n_i of zone i and consider the transition fex_i with rate ρ that leads to symbolic marking $\mathbf{Ms}_1$. Let us consider processes, say Π' in $\mathbf{Ms}'$ matching this block. There are several cases to be examined.

• **Case 2.1: The processes of Π' are beyond zone i.** $\mathbf{Ms}_1 \cdot geint_i = \mathbf{Ms} \cdot geint_i + s_i$, thus we have to check whether the condition $\mathbf{Ms}' \cdot geint_i \geq \mathbf{Ms}_1 \cdot geint_i$ holds. Since Π is matched with Π', this implies $\mathbf{Ms}' \cdot geint_i > \mathbf{Ms} \cdot geint_i$ and since $\mathbf{Ms}' \cdot geint_i - \mathbf{Ms} \cdot geint_i$ is a multiple of s_i, this implies that $\mathbf{Ms}' \cdot geint_i \geq \mathbf{Ms} \cdot geint_i + s_i = \mathbf{Ms}_1 \cdot geint_i$. Condition (C2) still holds with the same R_i's.

• **Case 2.2: Π' is a block in zone i.** This implies that Π and Π' are matched by relation R_i. In addition, since all the processes of Π are in location n_i, all processes of Π' are also in location n_i. There are two subcases to be considered.

•• **Case 2.2.1: fex_i is firable (with Π') in $\mathbf{Ms}'$ leading to $\mathbf{Ms}'_1$ with rate μ'.** Due to the requirements on the net $\mu' \geq \mu$. Adding a self-loop around $\mathbf{Ms}$, we have to prove that $\mathbf{Ms} \leq \mathbf{Ms}'_1$ and $\mathbf{Ms}_1 \leq \mathbf{Ms}'_1$. The former relation comes from Lemma 1 and Lemma 2. Let us focus on the latter one. The single quantities that change for $(C1)$ are $\mathbf{Ms}_1 \cdot geint_i$ and $\mathbf{Ms}'_1 \cdot geint_i$ that are both incremented.

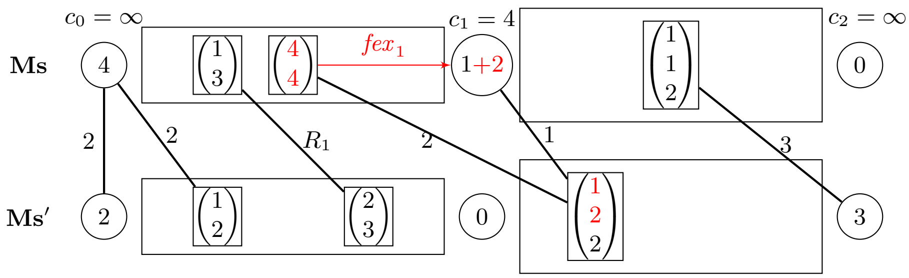


Fig. 16. Illustrating Case 2.1.

Thus $(C1)$ still holds. W.r.t. $(C2)$, Δ_i is decremented. So the multisets R_j are unchanged except R_i where the pair of blocks Π and Π' is deleted.

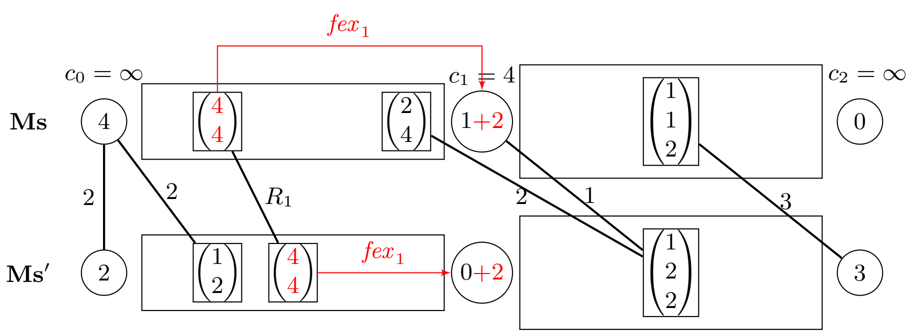


Fig. 17. Illustrating Case 2.2.1.

•• **Case 2.2.2: fex_i is not firable (with Π') in $\mathbf{Ms'}$.** Let us prove $\mathbf{Ms}_1 \leq \mathbf{Ms'}$. Since $\mathbf{Ms} \cdot Int_i < c_i \leq \mathbf{Ms'} \cdot Int_i$ and $\mathbf{Ms} \cdot gezone_i \leq \mathbf{Ms'} \cdot gezone_i$, one obtains $\mathbf{Ms} \cdot geint_i < \mathbf{Ms'} \cdot geint_i$. Since these two quantities are multiples of s_i, this implies that $\mathbf{Ms} \cdot geint_i + s_i \leq \mathbf{Ms'} \cdot geint_i$. The single quantity that changes for $(C1)$ is $\mathbf{Ms}_1 \cdot geint_i$ which is incremented by s_i. Thus $(C1)$ still holds. W.r.t. $(C2)$, Δ_i is decremented. So the multisets R_j are unchanged except R_i where the pair of blocks Π and Π' is deleted.

Case 3: A transition from an interface to a zone. Assume that there are at least s_{i+1} processes of $\mathbf{Ms}$ in the interface i, less than r_{i+1} blocks of processes of $\mathbf{Ms}$ in zone $i+1$. So transition fin^-_{i+1} (followed by transition fin^+_{i+1}) is firable with rate, say ρ. Denote $\mathbf{Ms}_1$ the symbolic marking reached by the sequence $fin^-_{i+1} fin^+_{i+1}$. There are two subcases to be considered.

• **Case 3.1: At least one process π of Ms in the interface i is matched with a process π' of Ms′ beyond interface i.** This implies

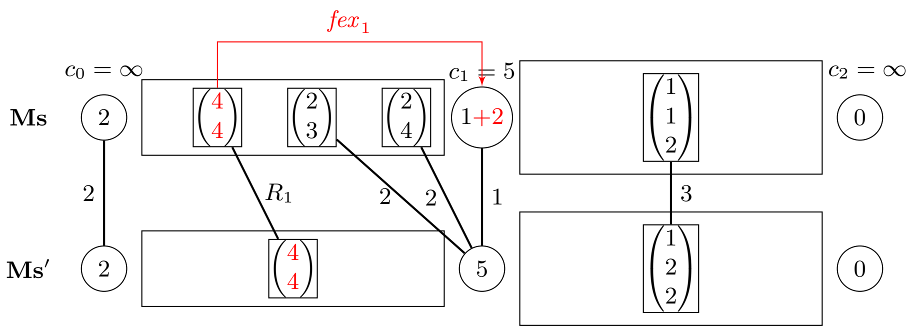


Fig. 18. Illustrating Case 2.2.2.

that $\mathbf{Ms}' \cdot gezone_{i+1} > \mathbf{Ms}' \cdot gezone_{i+1}$ which implies $\mathbf{Ms}' \cdot gezone_{i+1} \geq \mathbf{Ms}' \cdot gezone_{i+1} + s_{i+1}$. Thus at least s_{i+1} processes of **Ms** in the interface i are matched with processes of $\mathbf{Ms}'$ beyond interface i. We only need to prove that $\mathbf{Ms}_1 \leq \mathbf{Ms}$. Condition $(C1)$ is still satisfied since $\mathbf{Ms}_1 \cdot gezone_{i+1}$ is incremented by s_{i+1} and $\mathbf{Ms}' \cdot gezone_{i+1} \geq \mathbf{Ms} \cdot gezone_{i+1} + s_{i+1}$. Verifying condition $(C2)$ requires to examine two subcases.

•• Case 3.1.1: There are at least s_{i+1} matching processes in $\mathbf{Ms}'$ beyond zone $i+1$. This implies that $\mathbf{Ms}' \cdot geint_{i+1} \geq \mathbf{Ms} \cdot gezone_{i+1} + s_{i+1}$. So Δ_{i+1} w.r.t. the pairs $(\mathbf{Ms}, \mathbf{Ms}')$ and $(\mathbf{Ms}_1, \mathbf{Ms}')$ is null. Thus condition $(C2)$ holds with the same R_i's.

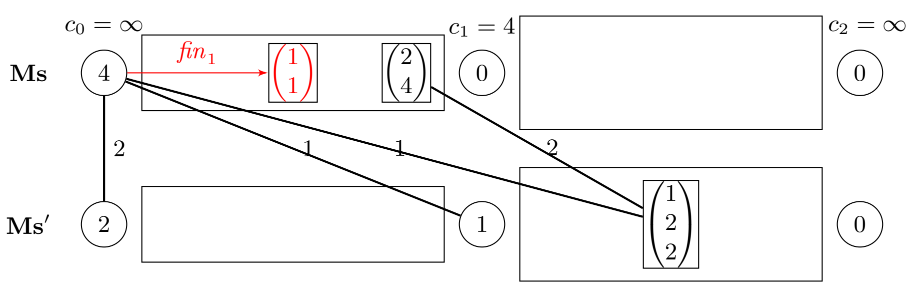


Fig. 19. Illustrating Case 3.1.1.

•• Case 3.1.2: There are less than s_{i+1} matching processes of $\mathbf{Ms}'$ beyond zone $i+1$. We claim that, in this case, there is no matching process in $\mathbf{Ms}'$ beyond zone $i+1$. Indeed $\mathbf{Ms}' \cdot geint_{i+1}$ and $\mathbf{Ms} \cdot gezone_{i+1}$ are multiples of s_{i+1}. If some process of **Ms** in interface i, would be matched with a process, since there are at least s_{i+1} processes of **Ms** in the interface i, s_{i+1} such processes could be matched. Thus the processes of $\mathbf{Ms}'$ beyond interface i that are matched with processes of $\mathbf{Ms}$ are in interface i, are in a block of zone $i+1$. So due the matching procedure, considering the first process in the interface that have been matched, we know that a (full) block of processes of $\mathbf{Ms}$ has been matched with s_{i+1} processes of $\mathbf{Ms}$ in the interface i. Thus Condition $(C2)$ holds with the same R_j's except R_{i+1} enlarged by the pair consisting of the new block of $\mathbf{Ms}_1$ and the matching block of $\mathbf{Ms}'$.

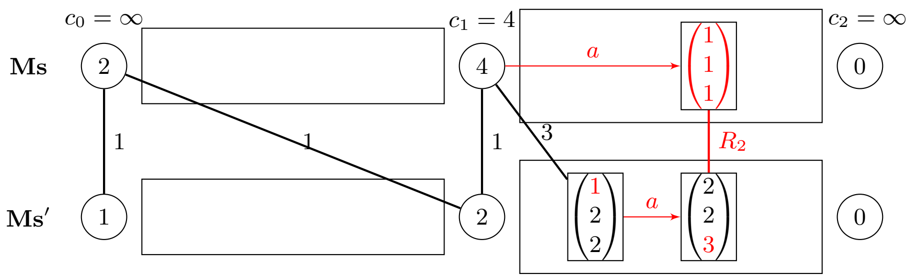


Fig. 20. Illustrating Case 3.1.2.

• **Case 3.2: All processes of Ms in the interface i are matched with processes of Ms′ in the interface i.** We first prove that there are less than r_{i+1} blocks of processes of $\mathbf{Ms}'$ in zone $i+1$. Observe that due to the assumption about matching, $\mathbf{Ms} \cdot gezone_{i+1} = \mathbf{Ms}' \cdot gezone_{i+1}$. We know that there is less than r_{i+1} blocks of processes of $\mathbf{Ms}$ in zone $i+1$. Thus:

$$\mathbf{Ms}' \cdot geint_{i+1} \geq \mathbf{Ms} \cdot geint_{i+1} > \mathbf{Ms} \cdot gezone_{i+1} - r_{i+1}s_{i+1} = \mathbf{Ms}' \cdot gezone_{i+1} - r_{i+1}s_{i+1}$$

which implies that there are less than r_{i+1} blocks of processes of $\mathbf{Ms}'$ in zone $i+1$. So the sequence of transitions $fin^{-}_{i+1} fin^{+}_{i+1}$ is also firable in $\mathbf{Ms}'$ with rate $\rho' \geq \rho$ since $\mathbf{Ms}' \cdot Int_i \geq \mathbf{Ms} \cdot Int_i$. Denote $\mathbf{Ms}'_1$ the symbolic marking that has been reached. By considering a self-loop for $\mathbf{Ms}$ with rate $\rho' - \rho$, one must show that $\mathbf{Ms}_1 \leq \mathbf{Ms}'_1$ and $\mathbf{Ms} \leq \mathbf{Ms}'_1$. Due to Lemma 1 and Lemma 2, we only have to prove that $\mathbf{Ms}_1 \leq \mathbf{Ms}'_1$. Condition $(C1)$ still holds since only $\mathbf{Ms}_1 \cdot gezone_{i+1}$ and $\mathbf{Ms}'_1 \cdot gezone_{i+1}$ are both incremented. Condition $(C2)$ also holds with the same R_j's except R_{i+1} enlarged with the pair of (identical) vectors corresponding to the new blocks in zone $i+1$ for $\mathbf{Ms}_1$ and $\mathbf{Ms}'_1$.

□

5 Stochastics bounds

This section is devoted to the comparison between stochastic processes modelled by nets. More precisely, we establish bounds between (1) nets which differ by their capacities on interfaces and zones and (2) nets which differ by their transition rates.

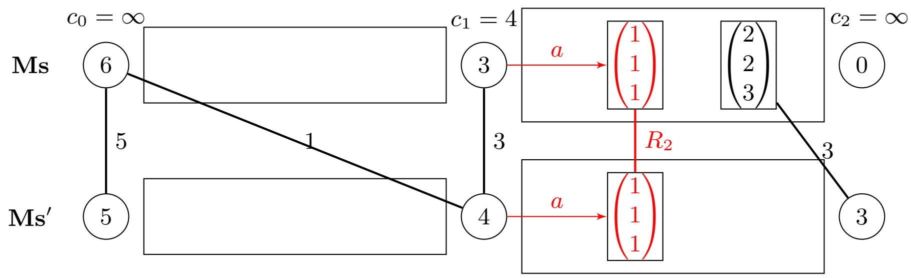


Fig. 21. Illustrating Case 3.2.

5.1 Stochastic MSN with different capacities

In order to establish bounds between MSN that differsby their capacities, we have to restric the class of MSN. Indeed intuitively, processes advance more freely when capacities are increased. However since a lower capacity can forbid a process to go backward, this intuition is only valid given the following restrictions.

Definition 12 (unidirectional MSN). *A unidirectional* MSN *is a* MSN *where the set of transitions T does not contain any bin_i^-, bex_i, bin_i^+ transitions.*

In the sequel of this subsection we compare unidirectional MSN $\mathcal{N}$ and $\widetilde{\mathcal{N}}$ that only differ by their capacities: $\forall i \leq \zeta$, $\tilde{c}_i \leq c_i$ and $\tilde{r}_i \leq r_i$. Observe that the state space of $\widetilde{\mathcal{N}}$ will be smaller that the one of $\mathcal{N}$. Thus in practice, $\mathcal{N}$ is the original model and $\widetilde{\mathcal{N}}$ is analysed in order to get bounds for performance indices of $\mathcal{N}$.

Theorem 3. *Consider the order of Definition 11 between symbolic markings of $\widetilde{\mathcal{N}}$ and symbolic markings of $\mathcal{N}$. Then this order defines a coupling relation.*

Proof. The proof of this theorem mimics the proof of Theorem 2. However there are two main differences: it only considers the case of forward transitions and it takes into account the difference between capacities.

Matching of processes is performed as in the part one of the proof of Theorem 2 since it does not involve capacities. Let us focus on the matching of transitions with a case per case analysis.

Case 1: A transition in a zone. The proof in this case is identical to the one of Case 1 of Theorem 2 as it does not involve capacities.

Case 2: Exiting a zone. Assume that all processes of a block, say $\widetilde{\Pi}$, of $\widetilde{\mathbf{Ms}}$ are in location n_i of zone i and consider the transition fex_i with rate ρ that leads to symbolic marking $\widetilde{\mathbf{Ms}}_1$. Let us consider processes, say Π in $\mathbf{Ms}$ matching this block. There are several cases to be examined.

• **Case 2.1: The processes of Π' are beyond zone i.** The proof of this case is identical to Case 2.1 in Theorem 2.

• **Case 2.2: Π is a block in zone i.** This implies that $\widetilde{\Pi}$ and Π are matched by relation R_i. In addition, since all the processes of $\widetilde{\Pi}$ are in location n_i, all processes of Π are also in location n_i. There are two subcases to be considered.

•• **Case 2.2.1: fex_i is firable in Ms.** The proof of this case is identical to Case 2.1.1 in Theorem 2.

•• **Case 2.2.2: fex_i is not firable (with Π) in Ms.** Let us prove $\widetilde{\mathbf{Ms}}_1 \leq \mathbf{Ms}$.
We have $\widetilde{\mathbf{Ms}} \cdot int_i \leq \tilde{c}_i - s_i \leq c_i - s_i < \mathbf{Ms} \cdot int_i$
and $\widetilde{\mathbf{Ms}} \cdot gezone_{i+1} \leq \mathbf{Ms} \cdot gezone_{i+1}$.
Therefore:
$\widetilde{\mathbf{Ms}} \cdot geint_i = \widetilde{\mathbf{Ms}} \cdot gezone_{i+1} + \widetilde{\mathbf{Ms}} \cdot int_i < \mathbf{Ms} \cdot gezone_{i+1} + \mathbf{Ms} \cdot int_i = \mathbf{Ms} \cdot geint_i$.
As $\widetilde{\mathbf{Ms}} \cdot geint_i$ and $\mathbf{Ms} \cdot geint_i$ are multiples of s_i, $\widetilde{\mathbf{Ms}} \cdot geint_i + s_i \leq \mathbf{Ms} \cdot geint_i$.
Finally, $\widetilde{\mathbf{Ms}}_1 \cdot geint_i = \widetilde{\mathbf{Ms}} \cdot geint_i + s_i \leq \mathbf{Ms} \cdot geint_i = \mathbf{Ms}_1 \cdot geint_i$

Case 3: A transition from an interface to a zone. Assume that there are at least s_{i+1} processes of $\widetilde{\mathbf{Ms}}$ in the interface i, strictly less than $\tilde{r}_{i+1}$ blocks of processes of $\widetilde{\mathbf{Ms}}$ in zone $i+1$. So transition fin^-_{i+1} (followed by transition fin^+_{i+1}) is firable with rate, say ρ. Denote $\widetilde{\mathbf{Ms}}_1$ the symbolic marking reached by the sequence $fin^-_{i+1} fin^+_{i+1}$. There are two subcases to be considered.

• **Case 3.1: At least one process $\tilde{\pi}$ of $\widetilde{\mathbf{Ms}}$ in the interface i is matched with a process π of Ms beyond interface i.** The proof of this case is identical to Case 3.1 in Theorem 2.

• **Case 3.2: All processes of $\widetilde{\mathbf{Ms}}$ in the interface i are matched with processes of Ms in the interface i.** We first prove that there are strictly less than r_{i+1} blocks of processes of $\mathbf{Ms}$ in zone $i+1$. Observe that due to the assumption about matching, $\widetilde{\mathbf{Ms}} \cdot gezone_{i+1} = \mathbf{Ms} \cdot gezone_{i+1}$. We know that there is strictly less than $\tilde{r}_{i+1}$ blocks of processes of $\widetilde{\mathbf{Ms}}$ in zone $i+1$. Thus:

$$\mathbf{Ms} \cdot geint_{i+1} \geq \widetilde{\mathbf{Ms}} \cdot geint_{i+1} > \widetilde{\mathbf{Ms}} \cdot gezone_{i+1} - \tilde{r}_{i+1} s_{i+1} \geq \mathbf{Ms} \cdot gezone_{i+1} - r_{i+1} s_{i+1}$$

which implies that there are strictly less than r_{i+1} blocks of processes of $\mathbf{Ms}$ in zone $i+1$. So the sequence of transitions $fin^-_{i+1} fin^+_{i+1}$ is also firable in $\mathbf{Ms}$ with rate $\rho' \geq \rho$ since $\mathbf{Ms} \cdot Int_i \geq \widetilde{\mathbf{Ms}} \cdot Int_i$. Denote $\mathbf{Ms}_1$ the symbolic marking that has been reached. By considering a self-loop for $\widetilde{\mathbf{Ms}}$ with rate $\rho' - \rho$, one must show that $\widetilde{\mathbf{Ms}}_1 \leq \mathbf{Ms}_1$ and $\widetilde{\mathbf{Ms}} \leq \mathbf{Ms}_1$. Due to Lemma 1 and Lemma 2, we only have to prove that $\widetilde{\mathbf{Ms}}_1 \leq \mathbf{Ms}_1$. Condition $(C1)$ still holds since only $\widetilde{\mathbf{Ms}}_1 \cdot gezone_{i+1}$ and $\mathbf{Ms}_1 \cdot gezone_{i+1}$ are both incremented. Condition $(C2)$ also holds with the same R_j's except R_{i+1} enlarged with the pair of (identical) vectors corresponding to the new blocks in zone $i+1$ for $\widetilde{\mathbf{Ms}}_1$ and $\mathbf{Ms}_1$.

□

5.2 Stochastic MSN with static subclasses

In MSN there is no static subclasses. However most of systems include different kinds of processes or resources: for instance, machines of a FMS may have different characteristics while ensuring the same function. In order to take into account this feature, we introduce the class of pre-monotonic nets (PMN).

Definition 13 (Pre-Monotonic Net (PMN)). *A* PMN $\mathcal{N} = (\Sigma, \mathcal{C}, \delta, P, T, \mathbf{prio}, cd, \mathbf{Guard}, \mathbf{Pre}, \mathbf{Post}, w)$ *is defined as a* MSN *with the exception of:*

- *Proc is partitioned into static subclasses with np the number of static subclasses in Proc:* $Proc = \bigcup_{j=1}^{np} Proc_j$*;*
- δ *is not required to fulfill the monotonic conditions.*

In order to specify the stochastic behaviour of a PMN, we introduce more refined counters than the ones of a MSN. Let m be a marking.

- $m \cdot int_{i,j}$ is the number of $Proc_j$ processes in interface i;
- $m \cdot geint_{i,j}$ is the number of $Proc_j$ processes in interfaces k for $k \geq i$ and zones k for $j > i$;
- $m \cdot gezone_{i,j}$ is the number of $Proc_j$ processes in interfaces k and zones k for $k \geq i$ (thus $m \cdot gezone_{i+1,j} = m \cdot geint_{i,j} - m \cdot int_{i,j}$).

We are now in position to provide a stochastic behaviour to PMN.

Definition 14 (Stochastic PMN). *A stochastic pre-monotonic net is a* PMN $\mathcal{N}$ *where the weight of transitions is defined as follows.*

- *For all immediate transitions* $fint_i^+, bint_i^+$ *the weight is arbitrary;*
- *Let* $t \in \{t_{i,j,a} \mid 1 \leq i \leq \zeta, 1 \leq j \leq n_i, a \in \Sigma, \delta(i,j,a) \text{ is defined}\} \cup \{fex_i, bex_i \mid 1 \leq i \leq \zeta\}$*. The counting vector* f_t *is defined by counters* $\{geint_{i,j}\}_{1\leq i\leq\zeta, 1\leq j\leq np}, \{gezone_{i,j}\}_{0\leq i\leq\zeta, 1\leq j\leq np}$ *;* g_t *is a non decreasing (resp. non increasing) function w.r.t. any counter when t is some* $t_{i,k,a}$ *with* $a \in \Sigma_f$ *(resp.* $a \in \Sigma_b$*) or some* fex_i *(resp.* bex_i*).*
- *Let* $1 \leq i \leq \zeta$ *and* $t \in \{fin_i^-, bin_i^-\}$*. Then counting vector* f_t *is defined by the counters* $\{geint_{j,k}\}_{1\leq j\leq\zeta, 1\leq k\leq np}, \{gezone_{j,k}\}_{0\leq j\leq\zeta, 1\leq k\leq np}, int_i$ *;* g_t *is non decreasing (resp. non increasing) function w.r.t. any counter when* $t = fin_i^-$ *(resp.* bin_i^-*).*

Our aim is to substitute a pre-monotonic net $\mathcal{N}$ by a monotonic one $\overline{\mathcal{N}}$ and get bounds on performance indices of $\mathcal{N}$ by analysis of $\overline{\mathcal{N}}$. In order to this, we define a natural mapping *abs* from counters of a PMN to counters of a MSN. Let ***cpt*** be a counter vector of a PMN, then $abs(\boldsymbol{cpt})$ is defined by:

- $m \cdot int_i = \sum_j m \cdot int_{i,j}$;
- $m \cdot geint_i = \sum_j m \cdot geint_{i,j}$;
- $m \cdot gezone_i = \sum_j m \cdot gezone_{i,j}$.

Definition 15. *Let* $\mathcal{N} = (\Sigma, \mathcal{C}, \delta, P, T, \mathbf{prio}, cd, \mathbf{Guard}, \mathbf{Pre}, \mathbf{Post}, w)$ *be a pre-monotonic stochastic net. Then the stochastic* MSN *associated with* $\mathcal{N}$*,* $\overline{\mathcal{N}} = (\Sigma, \overline{\mathcal{C}}, \overline{\delta}, P, T, \mathbf{prio}, cd, \mathbf{Guard}, \mathbf{Pre}, \mathbf{Post}, \overline{w})$ *is defined as follows:*

- $\overline{\mathcal{C}} = \{Proc\} \cup \{Sync_i\}_{1\leq i\leq\zeta}$ *with no static subclasses.*

– *For all* $1 \leq i \leq \zeta$ *and* $a \in \Sigma_f$ *:*

$$\overline{\delta}(i,j,a) = \max_{k \leq j}\{\delta(i,k,a) \mid \delta(i,k,a) \textit{ is defined } \wedge \delta(i,k,a) \geq j\}$$

when the set $\{k \mid k \leq j \wedge \delta(i,k,a) \textit{ is defined } \wedge \delta(i,k,a) \geq j\}$ *is non empty and is undefined otherwise;*

– *For all* $1 \leq i \leq \zeta$ *and* $a \in \Sigma_b$ *:*

$$\begin{aligned}\overline{\delta}(i,j,a) = \quad & \max_{k \leq j}(\{\delta(i,k,a) \mid \delta(i,k,a) \textit{ is defined } \wedge \delta(i,k,a) \leq j\} \\ & \qquad \cup\{k \mid \delta(i,k,a) \textit{ is undefined }\})\end{aligned}$$

when the set $\{\delta(i,k,a) \mid \delta(i,k,a) \textit{ is defined } \wedge \delta(i,k,a) \leq j\}$ *is non empty and is undefined otherwise;*

– *For* $fint_i^+$ *and* $bint_i^+$*, the weight function* $\overline{w}$ *is* 1*. Otherwise, it is defined as* $\overline{g}_t \circ \overline{f}_t$ *where* $\overline{f}_t$ *is defined as for the monotonic case. The function* $\overline{g_t}$ *is defined for all* $1 \leq i \leq \zeta$ *by:*

$$\overline{g}_{t_{i,j,a}}(\boldsymbol{cpt}) = \max_{k \leq j} \max_{\boldsymbol{cpt'}}\{g_{t_{i,k,a}}(\boldsymbol{cpt'}) \mid abs(\boldsymbol{cpt'}) = \boldsymbol{cpt} \wedge \overline{\delta}(i,k,a) > j\}$$

when $a \in \Sigma_f$ *and by:*

$$\overline{g}_{t_{i,j,a}}(\boldsymbol{cpt}) = \min_{k \geq j} \min_{\boldsymbol{cpt'}}\{g_{t_{i,k,a}}(\boldsymbol{cpt'}) \mid abs(\boldsymbol{cpt'}) = \boldsymbol{cpt} \wedge \delta(i,k,a) < j\}$$

when $a \in \Sigma_b$*.*

By construction, $\overline{\mathcal{N}}$ is a MSN. In order to establish the coupling relation between markings of $\mathcal{N}$ and markings $\overline{\mathcal{N}}$ we cannot use directly the order of Definition 11 since it applies to symbolic markings. However it can be straightforwardly adapted as follows. Let m be a marking of $\mathcal{N}$ and $\overline{m}$ be a marking of $\overline{\mathcal{N}}$. When forgetting the static subclasses one can associate with m a symbolic marking $\mathbf{Ms}$ of $\overline{\mathcal{N}}$. Let us denote $\overline{\mathbf{Ms}}$ the symbolic marking associated with $\overline{m}$. Then $m \leq \overline{m}$ if $\mathbf{Ms} \leq \overline{\mathbf{Ms}}$.

Theorem 4. *The relation defined above between markings of* $\mathcal{N}$ *and* $\overline{\mathcal{N}}$ *is a coupling.*

Proof. Due to Theorem 2, the relation $\leq$ is a coupling between markings of $\overline{\mathcal{N}}$. Thus it is enough to compare transitions outgoing from a single marking m in $\mathcal{N}$ and $\overline{\mathcal{N}}$.
Then the result is a direct consequence of:

$$\begin{aligned}\delta(i,j,a) \leq & \quad \overline{\delta}(i,j,a) \\ a \in \Sigma_f \ b \text{ a binding of } t_{i,j,a} \Rightarrow \overline{w}_{t_{i,j,a}}(b,m) \geq & \quad w_{t_{i,j,a}}(m,b) \\ a \in \Sigma_b \ b \text{ a binding of } t_{i,j,a} \Rightarrow \overline{w}_{t_{i,j,a}}(m,b) \leq & \quad \overline{w}_{t_{i,j,a}}(b,m)\end{aligned}$$

since every forward (resp. backward) transition in $\overline{\mathcal{N}}$ can be matched with the (possible) corresponding transition in $\mathcal{N}$ and a loop.

Fig. 22. Open monotonic symmetric nets.

6 Extensions

6.1 Open systems

Net structure. Here we want to express open systems where processes can be dynamically created and when finishing their tasks are killed. This extension can be done in a natural and simple way. First, for the formalism point of view, we add an input transition *enter* that has no input places and a single output place Int_0 as illustrated in Figure 22. We also add an immediate transition *exit* that consumes the tokens of the last interface place Int_ζ. Thus any tangible reachable marking does not contain tokens in Int_ζ.

Coupling relation. However the coupling relation has to be adapted. Observe first that given two symbolic reachable markings $\mathbf{Ms}$ and $\mathbf{Ms}'$, the number of processes that are present, i.e. $\mathbf{Ms}.geint_0$ and $\mathbf{Ms}'.geint_0$, may be different. W.r.t the intended coupling relation, the symbolic marking that has less processes should be more advanced. Since we want to reuse the previous definition, we simply add the *missing processes* to place Int_ζ. This leads to the following definition.

Definition 16. *Let $\mathcal{N}$ be an open* MSN *and* $\mathbf{Ms}, \mathbf{Ms}'$ *be two symbolic reachable markings of* $\mathcal{N}$*. Then* $\mathbf{Ms} \leq \mathbf{Ms}'$ *if:*

- $\mathbf{Ms}.geint_0 \geq \mathbf{Ms}'.geint_0$;
- *Let* $\mathbf{Ms}^*$ *be equal to* $\mathbf{Ms}'$ *except for place* Int_ζ: $\mathbf{Ms}^*(Int_\zeta) = \mathbf{Ms}.geint_0 - \mathbf{Ms}'.geint_0$*. Then* $\mathbf{Ms} \leq \mathbf{Ms}^*$ *w.r.t. Definition 11.*

Firing rates. We still allow rate dependencies for "internal" transitions of the open net. However they cannot be defined as previously done. First we introduce new abbreviations: $\mathbf{Ms}.leint_i$ (resp. $\mathbf{Ms}.lezone_i$) representing the number of processes avanced at most up to Interface i (resp. Zone i). Then a forward (resp. backward) transition may depend in a non increasing (resp. non decreasing) way of these parameters. The dependency on $\mathbf{Ms}.int_i$ is still valid for transitions fin_{i+1} and bin_i. Transition *enter* is handled like a backward transition since it "delays" the system to be empty.

Example 10 (Tandem queue). Let us consider the tandem queue (presented on the left of Figure 23) already discussed in section 2. The open monotonic symmetric net is presented on the right of Figure 23. It consists in three interfaces (and no zone) where the queues correspond to the first interfaces. Since the net

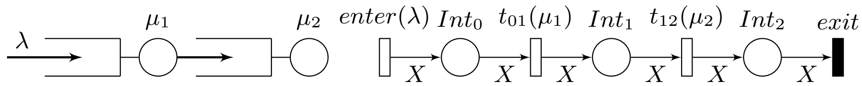


Fig. 23. On the left: a tandem queue. On the right: the corresponding net.

does not include synchronization, it could be transformed into an ordinary net. Interestingly, we can express most of the variants for queues. The second queue may have a finite capacity; more generally, in case of several successive queues all but the first queue may have a finite capacity. Thanks to the possible dependency on the marking of interfaces, we are able to express the standard service policies (single-server, multiple-server, infinite-server).

Performance indices. Using the coupling relation for open monotonic symmetric nets, several useful performance indices can be bounded among them:

- the busy period which is the time between the entrance of a process when there is not already other ones in the net and the departure of a process letting no process in the net;
- the number of processes in the system both in transient and steady-state context;
- the mean completion time of a process in the steady-state context.

6.2 Closed systems

Net structure. Adapting our framework for closed systems is more difficult due to the unwanted interaction between processes that have not achieved the same number of "rounds" of the system. In order to avoid this problem. We introduce a new class of colours: *Round* which is nothing else that the set of integers. With every process is associated its current round initially set to 0. There is an additional transition *loop* that moves a process from Interface ζ to Interface 0 incrementing its round as presented in Figure 24. Furthermore all synchronized items are duplicated by rounds and synchronization is only allowed between processes with same round. Expression $\#Block_i(All, R)$ returns the number of available blocks with round R in place $Block_i$.

Firing rates. We still allow rate dependencies for "internal" transitions of the closed net. However they must explicitely refer to the round like $\mathbf{Ms}.geint_{i,r}$ whose meaning is the number of processes with round $r' > r$ plus the number of processes with round r that are advanced at least to Interface i. Transition *loop* is handled as a forward transition.

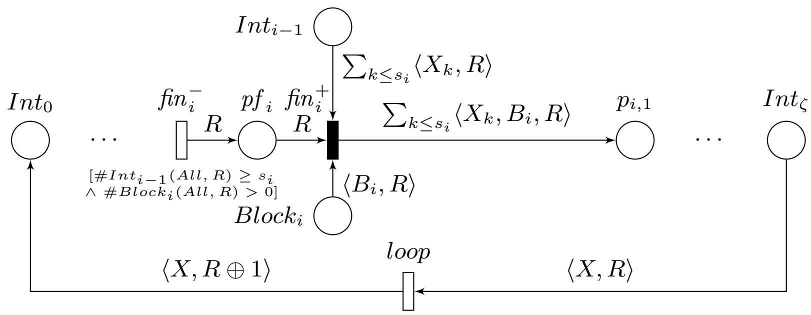


Fig. 24. Closed monotonic symmetric nets.

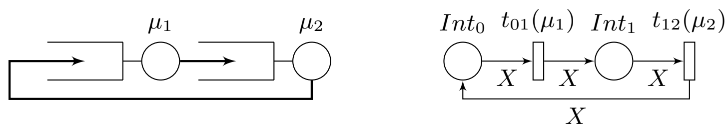


Fig. 25. From queuing networks to closed monotonic symmetric nets.

Performance indices. Using the coupling relation for closed monotonic symmetric nets, several useful performance indices can be bounded among them:

- the time for all processes to enter round r;
- the time for at least one process to enter round r;
- the throughput of the system, i.e. the frequency of *loop* firings.

Example 11 (Closed tandem queue). In figure 25, we have shown how a closed tandem queue (discussed in section 2) can be modelled by a closed monotonic symmetric net. Since there is neither synchronization nor resource sharing between processes, there is no need to memorize the current round of the processes.

7 Conclusion

In this work, we have developed a framework for which a bounding model can be built automatically. This framework is enough powerful to express resource allocations and synchronizations between processes. The modeling of a flexible manufacturing system has shown its practical interest for industrial case studies. We have also established that standard bounding models for queuing systems can be easily expressed within this framework.

Since our formalism is a particular case of stochastic symmetric nets, we plane to integrate our technique into GreatSPN [7]. Furthermore, it could be also used in Cosmos [2], a statistical model checker, for its rare event method which is based on the construction of a reduced model to be numerically solved in order to bias the sampling of transitions of the original model [3].

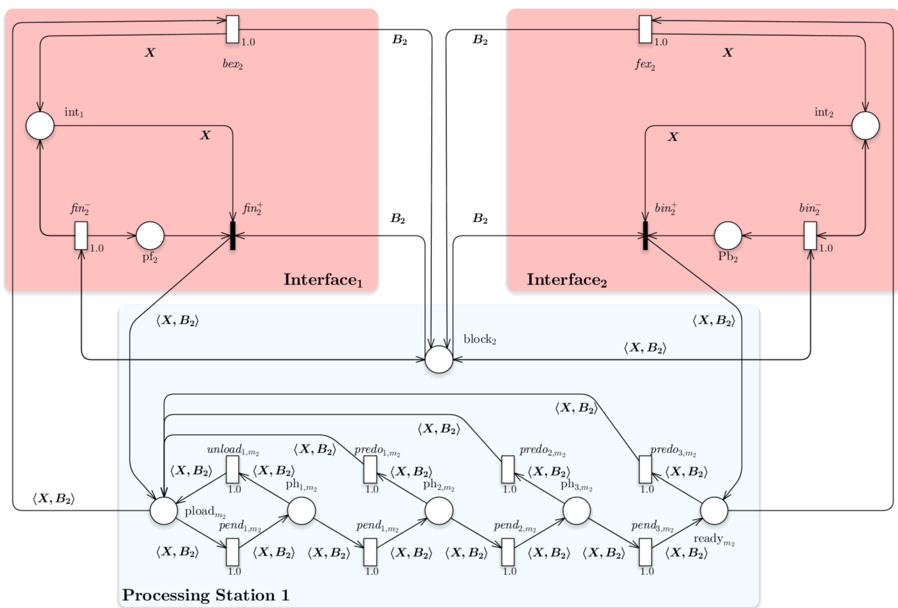
References

1. O. Abu-Amsha and J.-M. Vincent. An algorithm to bound functionals of Markov chains with large state space. In *4th INFORMS Conference on Telecommunications*, number 3.2, pages 3–2, 1998.
2. P. Ballarini, H. Djafri, M. Duflot, S. Haddad, and N. Pekergin. Cosmos: A statistical model checker for the hybrid automata stochastic logic. In *Quantitative Evaluation of Systems (QEST), 2011 Eighth International Conference on*, pages 143–144. IEEE, 2011.
3. B. Barbot, S. Haddad, and C. Picaronny. Coupling and importance sampling for statistical model checking. In C. Flanagan, B. K. Flanagan, and B. König, editors, *TACAS*, volume 7214 of *Lecture Notes in Computer Science*, pages 331–346. Springer, 2012.
4. H. Castel, L. Mokdad, and N. Pekergin. Stochastic bounds applied to the end to end qos in communication systems. In *15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2007), October 24-26, 2007, Istanbul, Turkey*, pages 374–380. IEEE Computer Society, 2007.
5. G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. Stochastic well-formed colored nets and symmetric modeling applications. *IEEE Transactions on Computers*, 42(11):1343–1360, Nov. 1993.
6. G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. A symbolic reachability graph for coloured Petri nets. *Theoretical Computer Science*, 176(1-2):39–65, Apr. 1997.
7. G. Chiola, G. Franceschinis, R. Gaeta, and M. Ribaudo. Greatspn 1.7: Graphical editor and analyzer for timed and stochastic Petri nets. *Perform. Eval.*, 24(1-2):47–68, 1995.
8. J. Fourneau, M. Lecoz, and F. Quessette. Algorithms for an irreducible and lumpable strong stochastic bound. *Linear Algebra and its Applications*, 386:167–185, 2004.
9. J. Fourneau, L. Mokdad, and N. Pekergin. Stochastic bounds for performance evaluation of web services. *Concurrency and Computation: Practice and Experience*, 22(10):1286–1307, 2010.
10. J. Fourneau, L. Mokdad, N. Pekergin, and S. Youcef. Stochastic bounds for loss rates. In *Third Advanced International Conference on Telecommunications (AICT 2007), May 13-19, 2007, Mauritius*, page 9. IEEE Computer Society, 2007.
11. J.-M. Fourneau and N. Pekergin. An algorithmic approach to stochastic bounds. In M. Calzarossa and S. Tucci, editors, *Performance Evaluation of Complex Systems: Techniques and Tools*, volume 2459 of *Lecture Notes in Computer Science*, pages 64–88. Springer, 2002.
12. S. Haddad, L. Mokdad, and S. Youcef. Bounding models families for performance evaluation in composite web services. *J. Comput. Science*, 4(4):232–241, 2013.
13. S. Haddad and P. Moreaux. Sub-stochastic matrix analysis for bounds computation - theoretical results. *European Journal of Operational Research*, 176(2):999–1015, 2007.
14. T. Lindvall. *Lectures on the coupling method*. Dover, 2002.
15. L. Mokdad and H. Castel-Taleb. Stochastic comparisons: A methodology for the performance evaluation of fixed and mobile networks. *Computer Communications*, 31(17):3894–3904, 2008.
16. B. Plateau. On the stochastic structure of parallelism and synchronization models for distributed algorithms. In *Proceedings of the 1985 ACM SIGMETRICS Conference*

on Measurement and Modeling of Computer Systems, SIGMETRICS '85, pages 147–154, New York, NY, USA, 1985. ACM.

A An SSN model for FMS

In this appendix we complete the description of the SSN model for FMS. The SSN sub-models corresponding to the remaining FMS zones are presented while the whole model is the composition of these sub-models by superposition of places with identical names.

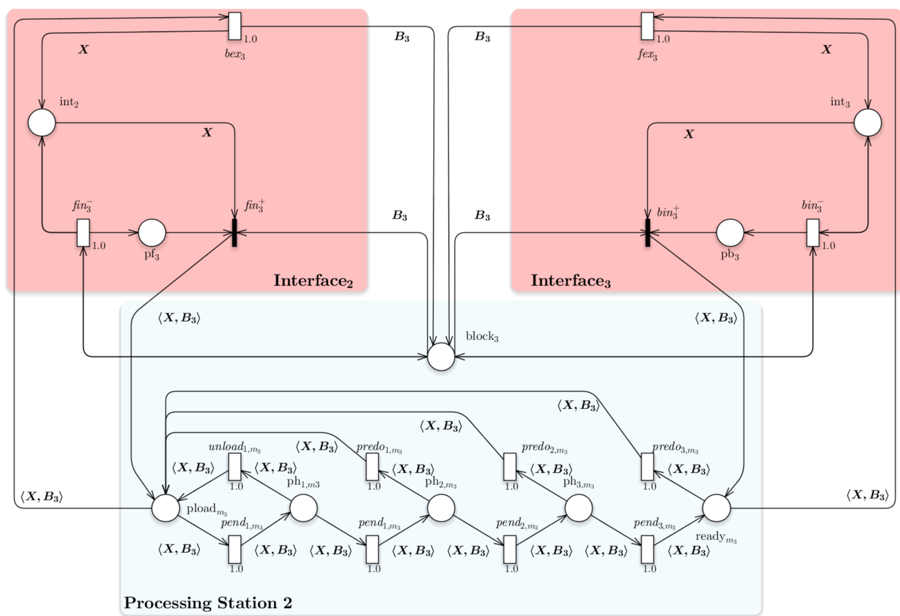


Initial marking: $m_0 = All.block_2$

Fig. 26. The first manufacturing station with a single machine.

Fig. 26 shows the SSN sub-model describing the second FMS zone which contains the first processing station machine (i.e. sub-net in blue box) and its input and output buffers (i.e. sub-nets in the red boxes). This sub-model is similar to the load part but a different number of phases are modeled, so that the sub-net composed by place ph_{i,m_1}, transitions $pend_{i,m_1}$ and $predo_{i,m_1}$ is instanciated three times. The initial marking for this model assumes all the processing station machines initially idle idle (i.e. $All.block_1$).

The SSN sub-model describing the third FMS zone is reported in Fig. 27. This sub-model is identical to the previous one, so the second processing station machine (i.e. sub-net in blue box) and its input and output buffers (i.e. sub-nets in the red boxes) are modeled as those in second zone. As before the processing station machines are initially idle (i.e. $All.block_2$).



Initial marking: $m_0 = All.block_3$

Fig. 27. The second manufacturing station including m machines.